REFORM: Error-Aware Few-Shot Knowledge Graph Completion

Song Wang
University of Virginia
sw3wv@virginia.edu

Xiao Huang
Hong Kong Polytechnic University
xiaohuang@comp.polyu.edu.hk

Chen Chen
University of Virginia
chenannie45@gmail.com

Liang Wu
Linkedin
liawu@linkedin.com

Jundong Li
University of Virginia
jundong@virginia.edu

ABSTRACT

Knowledge graphs (KGs) are of great importance in various artificial intelligence systems, such as question answering, relation extraction, and recommendation. Nevertheless, most real-world KGs are highly incomplete, with many missing relations between entities. To discover new triples (i.e., head entity, relation, tail entity), many KG completion algorithms have been proposed in recent years. However, a vast majority of existing studies often require a large number of training triples for each relation, which contradicts the fact that the frequency distribution of relations in KGs often follows a long tail distribution, meaning a majority of relations have only very few triples. Meanwhile, since most existing large-scale KGs are constructed automatically by extracting information from crowd-sourcing data using heuristic algorithms, plenty of errors could be inevitably incorporated due to the lack of human verification, which greatly reduces the performance for KG completion. To tackle the aforementioned issues, in this paper, we study a novel problem of error-aware few-shot KG completion and present a principled KG completion framework REFORM. Specifically, we formulate the problem under the few-shot learning framework, and our goal is to accumulate meta-knowledge across different meta-tasks and generalize the accumulated knowledge to the meta-test task for error-aware few-shot KG completion. To address the associated challenges resulting from insufficient training samples and inevitable errors, we propose three essential modules *neighbor encoder, cross-relation aggregation*, and *error mitigation* in each meta-task. Extensive experiments on three widely used KG datasets demonstrate the superiority of the proposed framework REFORM over competitive baseline methods.

CCS CONCEPTS

• **Computing methodologies → Knowledge representation and reasoning**.

KEYWORDS

Knowledge graphs, few-shot learning, graph neural networks

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11...$15.00
https://doi.org/10.1145/3459637.3482470

ACM Reference Format:
Song Wang, Xiao Huang, Chen Chen, Liang Wu, and Jundong Li. 2021. REFORM: Error-Aware Few-Shot Knowledge Graph Completion. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3482470

1 INTRODUCTION

Knowledge graph (KG) completion [21, 35] is an increasingly essential and challenging task in various artificial intelligence systems. KGs organize a large collection of triples (i.e., head entity, relation, tail entity) as a special type of graph [20, 38], in which nodes and edges represent entities and relations, respectively. Generally, KGs provide us an efficient way to model general-purpose and domain-specific knowledge base with a variety of applications, such as question answering [14], relation extraction [12], and recommendation [40]. Nevertheless, most, if not all, of real-world KGs have a high degree of incompleteness [19], and many relations between entities are missing. KG completion (i.e., KG link prediction[2] and KG reasoning[25]) has been intensively studied [21]. Existing KG completion algorithms could roughly be categorized into two classes, i.e., logical rule-based and embedding-based. Predicting relations in few-shot classes is one of the key problems in KG completion [4, 31, 43, 45]. Many KG completion algorithms rely on a large number of instances (i.e., entity pairs) in each relation category as training samples [35, 41]. However, in real-world KGs, the frequency distribution of relation categories follows a long tail distribution [43]. A large proportion of relation categories are only associated with very few triples, referred to as *few-shot relations.* Completing few-shot relations is challenging because of the insufficient training samples in these relation categories.

Many few-shot KG completion models have been explored [4, 31, 43, 45] in the past few years, but they ignored the significant impact of errors in KGs. In practice, most existing large-scale KGs [3, 20, 26] are constructed by automatically extracting information from crowd-sourcing data using heuristic algorithms. However, plenty of errors are inevitably incorporated due to the lack of human verification [1, 20]. For example, NELL [3], a widely-adopted KG, has an estimated precision of 74% in the early version, which is far from desirable. While another popular KG, YAGO 4 [26], has a manually verified accuracy of 95% with some errors existing. Errors in few-shot relations, when serving as training samples, would have a considerable negative impact on the few-shot relation completion. Thus, in this paper, we investigate a novel problem of *error-aware few-shot KG completion.* The goal is to predict the missing relations for entity pairs that belong to few-shot relation categories while

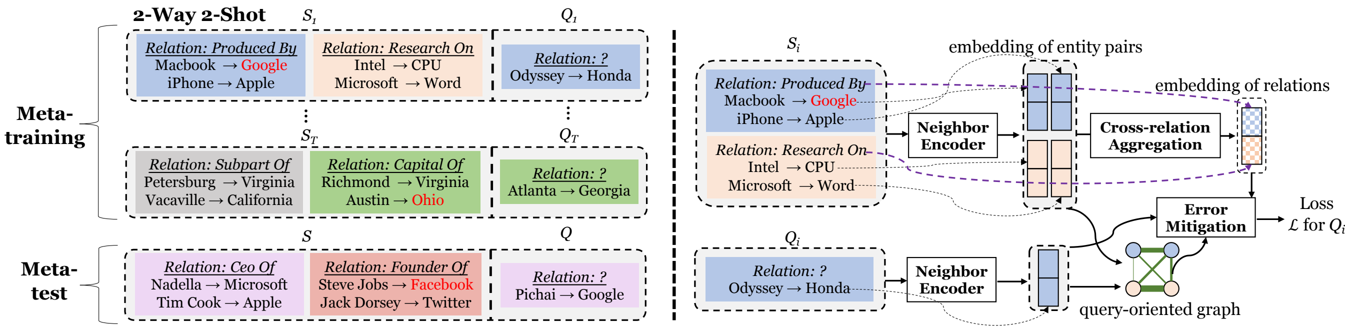

Figure 1: (Left) An illustration of the training process of the proposed framework REFORM, where red color denotes incorrect entities. (Right) An illustration of three essential components for each meta-learning task in REFORM.

significantly alleviating the impact of errors in the background (non-few-shot relations) and few-shot relations.

However, it remains a non-trivial task to perform *error-aware few-shot KG completion*. There are three major challenges as follows. Firstly, errors in KGs make it difficult to learn reliable entity/relation representations. Because of the unique data characteristics, many existing KG embedding algorithms (e.g., TransE [2] and ComplEx [35]) heavily rely on modeling triple structures. An incorrect entity or relation would significantly jeopardize the embedding expressiveness of its neighbors' embeddings. For example, if the head entity of a triple $(iPhone, ProducedBy, Apple)$ is polluted, the embedding representations of Apple will also be affected. Secondly, relations in KGs have semantic meanings and are more or less correlated with each other [46]. For instance, the relations *ProducedBy* and *Sell* are strongly correlated since these two relations may share many common entities (e.g., products). However, existing few-shot KG completion algorithms [43, 45] often assume that relation categories are independent of each other and ignore their inherent correlations. The third challenge is about the incorrect prediction resulting from incorrect triples of few-shot relations. Specifically, given a pair of entities, it is more easily to be misclassified to other relations if triples of the true few-shot relation are severely polluted.

To tackle the aforementioned challenges, we propose a novel eRror-awarE Few-shOt gRaph coMpletion framework—REFORM. Specifically, we formulate the problem under the prevalent few-shot learning framework, which accumulates meta-knowledge across a number of meta-training tasks and generalizes to a meta-testing task for error-aware few-shot KG completion. Essentially, to address the aforementioned challenges of few-shot KG completion, the proposed REFORM has three key components for each meta-task. Firstly, to alleviate the negative impact of erroneous neighbors, we design an attention-based neighbor encoder module to select more reliable neighbors for representation learning. Secondly, to take advantage of the correlations between different relations in the support set of each meta-task, we propose a cross-relation aggregation module to utilize the correlations of different relations for relation embedding via transformer encoders [36]. To alleviate the adverse impact brought by the errors at the prediction phase, we design an error mitigation module, which leverages graph convolution network (GCN) [17] to model the interactions between the query instance and the support instances, and generate confidence weights for different relations regarding each query so that the confidence of relations containing erroneous support instances will be reduced. In summary, our main contributions are three-fold:

- ***Problem Formulation:*** We discuss the limitations of existing research works on knowledge graph completion and make an initial investigation of a novel research problem—error-aware few shot knowledge graph completion.
- ***Algorithmic Design:*** We develop a novel few-shot learning framework for KG completion in the presence of errors. It consists of three essential modules for each meta-task: (1) an attention-based neighbor encoder module that reduces the effects of erroneous neighbors; (2) a cross-relation aggregation module based on transformer encoders which captures correlations between relations in the support set; and (3) a GCN-based error mitigation module that alleviates the impacts of erroneous triples at the prediction phase.
- ***Experimental Evaluation:*** We conduct extensive experiments on real-world knowledge graphs to validate the superiority of our proposed framework.

2 PROBLEM DEFINITION

Formally, a knowledge graph $\mathcal{G}$ consists of a number of triples $\{(h, r, t)\} \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where $\mathcal{E}$ and $\mathcal{R}$ are the entity set and relation set, respectively. Among these triples, many are erroneous (e.g., incorrect relation or incorrect head/tail entity). KG completion targets at learning from existing triples and predicting the unseen relation between a given entity pair: $(h, ?, t)$, or predicting an entity given a relation and another entity: $(?, r, t)$ and $(h, r, ?)$. In this work, we focus on the former scenario as it is closer to a real-world scenario where queries are mainly about the missing relations.

As mentioned previously, we aim to complete the triples for the few-shot relations, for which we use $\mathcal{R}_f$ to denote the set of few-shot relations that have incomplete triples (i.e., the triples whose relations are missing) and obviously $\mathcal{R}_f \subseteq \mathcal{R}$. Given the above, the studied problem of *error-aware few-shot KG completion* can be formally formulated as follows:

Definition 1. **Error-aware Few-shot KG Completion**: *Given (i) a knowledge graph $\mathcal{G}$ which includes a number of erroneous triples, (ii) a set of few-shot relations $\mathcal{R}_f$, and (iii) a query entity pair (h, t),* *our goal is to develop a machine learning model that can accurately predict the missing relation category r of the query entity pair from the few-shot relation categories $\mathcal{R}_f$.*

As each few-shot relation in $\mathcal{R}_f$ only consists of a limited amount of complete triples, we formulate the problem of *error-aware few-shot KG completion* under the prevalent few-shot learning paradigm, which is shown to be effective in many areas such as computer vision [34], natural language processing [44], and graph analysis [6]. Specifically, if $\mathcal{R}_f$ contains N different relation categories, then we can sample K complete triples for each few-shot relation category and formulate the problem as an N-way K-shot classification problem [37]. To mimic the real test environment for few-shot KG completion, we create a number of *meta-training* tasks from a disjoint relation set $\mathcal{R}_{f'} \subseteq \mathcal{R}$ and $\mathcal{R}_{f'} \cap \mathcal{R}_f = \emptyset$. Each meta-training task $\mathcal{T}_m$ consists of the support set $\mathcal{S}_m$ and the query set $\mathcal{Q}_m$:

$$\begin{aligned} \mathcal{S}_m &= \{(h_1^1, t_1^1, r_1), \ldots, (h_1^K, t_1^K, r_1), \ldots, \\ &\quad (h_N^1, t_N^1, r_N), \ldots, (h_N^K, t_N^K, r_N)\}, \\ \mathcal{Q}_m &= \{(\tilde{h}_1^1, \tilde{t}_1^1, \tilde{r}_1), \ldots, (\tilde{h}_1^M, \tilde{t}_1^M, \tilde{r}_1), \\ &\quad (\tilde{h}_N^1, \tilde{t}_N^1, \tilde{r}_N), \ldots, (\tilde{h}_N^M, \tilde{t}_N^M, \tilde{r}_N)\}. \end{aligned} \tag{1}$$

Here, $r_1, r_2, \ldots, r_N, \tilde{r}_1, \tilde{r}_2, \ldots, \tilde{r}_N \in \mathcal{R}_{f'}$. Also, (h_i^j, r_i, t_i^j) means that there is a relation r_i between the instance (i.e., entity pair) (h_i^j, t_i^j), which is the j-th instance of relation r_i in the support set. Similarly, $(\tilde{h}_i^j, \tilde{r}_i, \tilde{t}_i^j)$ denotes the j-th instance of relation $\tilde{r}_i$ in the query set $\mathcal{Q}_m$. The support set $\mathcal{S}_m$ contains K instances for each of N relations in $\mathcal{R}_{f'}$ while the query set $\mathcal{Q}_m$ consists of M query instances sampled from the remaining instances from each relation in $\mathcal{R}_{f'}$. With these, a meta-learning model is trained on a set of *meta-training* tasks $\mathcal{T}_{train} = \{\mathcal{T}_m\}_{m=1}^T$ to minimize the loss of predictions for each query set $\mathcal{Q}_m$. Through this process, the model gradually learns the meta-knowledge from these sampled *meta-training* tasks and thus can be effectively generalized to the *meta-test* task for few-shot KG completion. Specifically, the meta-task $\mathcal{T}_{test}$ also has a support set $\mathcal{S}$ and a query set $\mathcal{Q}$, and is defined in the same way as the *meta-training* task except that $\mathcal{Q}$ contains all remaining instances from each relation in $\mathcal{R}_f$. The support set $\mathcal{S}$ has K complete triples for each few-shot relation in $\mathcal{R}_f$ ($|\mathcal{R}_f| = N$), and our goal is to predict the missing few-shot relations for the instances in the query set $\mathcal{Q}$ by exploiting the support set $\mathcal{S}$ and the trained meta-learning model from T meta-training tasks.

3 PROPOSED FRAMEWORK

In this section, we introduce the overall structure of our proposed framework REFORM in detail. As illustrated in Figure 1, we formulate the *error-aware few-shot KG completion* problem under the prevalent N-way K-shot few-shot learning framework, which consists of T different meta-training tasks $\{\mathcal{S}_m, \mathcal{Q}_m\}_{m=1}^T$ and a meta-test task $\{\mathcal{S}, \mathcal{Q}\}$. Our goal is to learn the meta-knowledge across T different meta-training tasks and then generalize the learned knowledge to the meta-test task for few-shot KG completion. For each meta-task, our framework first designs a neighbor encoder module to generate robust embeddings for each entity pair by selecting more reliable neighboring instances with an attention mechanism. Then our framework employs a cross-relation aggregation module to better capture the correlations of relations in the support set. Moreover, the error mitigation module models the data-dependent confidence level for each relation in the support set w.r.t. each query instance. In this way, it can further reduce the false relation prediction risk of query instances induced by erroneous triples in the support set. Later on, the confidence scores are combined with the classification scores produced by a matching function and fed into the final classification layer. It should be noted that the neighbor encoder module and the error mitigation module both can alleviate the negative impact of errors for few-shot KG completion—the former module helps learn more reliable representations of entity pairs at the embedding phase while the latter module reduces the influence of errors at the relation prediction phase.

Next, we will elaborate on these three key modules that support error-aware few-shot KG completion.

3.1 Neighbor Encoder

Recent years have witnessed the great success of embedding based KG completion methods [2, 35, 41]. However, the existence of errors (e.g., incorrect relations or incorrect head/tail entities) may greatly jeopardize the expressiveness of the learned embeddings and consequently influence the performance of few-shot KG completion. The underlying reason is that if an entity or a relation is incorrect, the error will propagate to its immediate neighbors and neighbors that are multiple hops away during the embedding learning phase.

To generate more robust embedding representations in the presence of errors, we propose an attention-based neighbor encoder module. Specifically, given an instance (i.e., a pair of entities (h, t)), for each of its entities we generate a robust embedding representation by aggregating information from its neighboring relations and entities with an attention mechanism, such that the incorrect neighbors will be assigned lower weight values and correct neighbors will have higher weight values during the information aggregation process. For example, for an entity h and its neighboring entities $e_1, e_2, \ldots, e_m$ with corresponding relations $r_1, r_2, \ldots, r_m$ (m denotes the number of neighboring entities of h), we learn the contribution of e_i and r_i for learning the robust embedding of entity h via an attention mechanism of the bilinear form:

$$b_i = \frac{\mathbf{e}_h^T \mathbf{W}_n (\mathbf{r}_i \oplus \mathbf{e}_i)}{\sqrt{d_p}}, \tag{2}$$

where $\mathbf{e}_h \in \mathbb{R}^{d_p}$, $\mathbf{e}_i \in \mathbb{R}^{d_p}$, and $\mathbf{r}_i \in \mathbb{R}^{d_p}$ are the corresponding pretrained embeddings[1] for entities h, e_i and relation r_i, respectively, and d_p is the embedding size for entities and relations. Besides, $\oplus$ represents the concatenation operator between two vectors and $\mathbf{W}_n \in \mathbb{R}^{d_p \times 2d_p}$ is a trainable weight matrix. The above equation provides an attention weight for each neighbor, and then we normalize them using a softmax function to obtain the robust embedding of entity h as:

$$\alpha_i = \frac{\exp(b_i)}{\sum_{k=1}^m \exp(b_k)}, \tag{3}$$

$$\mathbf{h} = \mathbf{e}_h + \sum_{i=1}^m \alpha_i \left(\mathbf{W}_a(\mathbf{e}_i \oplus \mathbf{r}_i) + \mathbf{b}_a\right), \tag{4}$$

[1] In this paper, we employ TransE [2] to generate the pretrained embeddings.

where $\mathbf{h}$ is the encoded robust embedding of the head entity h. $\mathbf{W}_a \in \mathbb{R}^{d_p \times 2d_p}$ and $\mathbf{b}_a \in \mathbb{R}^{d_p}$ are a trainable weight matrix and a bias term for the robust embedding generation, respectively. In this way, we select the most reliable neighbors for the given entity h and aggregate neighbors' embeddings in a weighted sum format. Then the pretrained embedding of h is also included to incorporate the embedding representation of itself.

Similarly, we can also generate a robust embedding $\mathbf{t}$ for the tail entity t. Finally, we concatenate the embeddings of the head entity h and the tail entity t together to obtain a final embedding for the instance (i.e., entity pair) (h, r):

$$\mathbf{x} = \mathbf{h} \oplus \mathbf{t}, \tag{5}$$

where $\mathbf{x} \in \mathbb{R}^{2d_p}$ is the final robust embedding of this instance (h, t).

3.2 Cross-relation Aggregation

We have generated embeddings for instances (i.e., a pair of entities) through the previous neighbor encoder module, and now we focus on generating embeddings for relations in the support set. Different from image data [37] or text data [9], relations in KGs often come with semantic meanings and are more or less correlated with each other via certain levels of interactions [46]. For example, as mentioned previously, the relation *ProducedBy* and the relation *Sell* apparently have strong correlations as they encode similar semantic meanings and may share many common entities (e.g., products). When appearing in the same meta-task, these relations can benefit each other for learning their embeddings, which is essential as instances of each relation in the meta-task are often very limited. However, a vast majority of existing few-shot KG completion models [43, 45] do not fully take advantage of such correlations and may lead to suboptimal KG completion results.

Therefore, for each meta-task in our framework, we exploit the inherent correlations among different relations in the support set and utilize the correlations for relation embedding learning. More specifically, we incorporate not only the information from the support instances (i.e., entity pairs in the support set) of the same relation category, but also the information regarding the relation correlations in the support set. Toward this goal, we propose a cross-relation aggregation module based on transformer encoders [36] to capture these two types of information. Generally speaking, the transformer encoders make each input embedding attend to all other input embeddings based on a multi-head attention mechanism and thus have a strong ability in modeling interactions between different input embeddings.

Firstly, to incorporate the information from support instances of the same relation type, we apply a transformer encoder to produce an intermediate embedding for each relation category in the support set as follows:

$$\hat{\mathbf{r}}_i = \text{mean}\left(\text{Transformer}(\mathbf{x}_i^1, \mathbf{x}_i^2, \ldots, \mathbf{x}_i^K)\right), \tag{6}$$

where $\mathbf{x}_i^j$ denotes the embedding of the j-th support instance of the i-th relation in the support set, which is generated by the neighbor encoder module in the previous subsection. K is the number of support instances for each relation category and $\hat{\mathbf{r}}_i$ is the intermediate embedding of the i-th relation. By applying the transformer encoder for all the instances of the same relation category and calculating the mean, it enables us to obtain a comprehensive understanding for all the instances of the i-th relation in the support set.

Secondly, to further capture the inherent correlations among different relations in the support set, we apply another transformer encoder based on the intermediate embedding representations:

$$\mathbf{r}_a = \text{mean}\left(\text{Transformer}(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2, \ldots, \hat{\mathbf{r}}_N)\right), \tag{7}$$

where N denotes the number of relations for each relation category in the support set. Here, $\mathbf{r}_a$ can be interpreted as a context embedding which encodes the inherent interactions among different relation categories in the same support set.

Then an element-wise product operation is applied to obtain the final embedding of the i-th relation in the support set:

$$\mathbf{r}_i = \hat{\mathbf{r}}_i \odot \mathbf{r}_a, \tag{8}$$

where $\odot$ denotes the element-wise product operation. In this way, we can fuse the information regarding the instances of the i-th relation and the information regarding relation interactions in the same support set more synergistically.

3.3 Error Mitigation

As mentioned before, errors widely exist in KGs [3, 13]. Under the few-shot learning framework [37], the number of support instances in each meta-task is very limited, and thus a small number of errors in the support set can greatly affect the accumulation of meta-knowledge and consequently influence the KG completion performance [9]. Although the previous neighbor encoder module helps reduce the impact of errors at the embedding phase, incorrect relation predictions may still occur for query instances (i.e., entity pairs in the query set) due to inevitable errors in the support set.

To reduce the adverse impact of inevitable errors, we leverage graph convolution network (GCN) [17] to generate confidence weights of different relations for each query instance. The intuition is that due to inevitable errors in the support set, the confidential levels of different relations are different. For example, if a relation contains a large number of erroneous support instances, the confidence that a specific query instance belonging to this relation should also be reduced, which means this relation is not reliable for the relation prediction. A natural way to solve this problem is to assign a confidence weight for each relation based on the information of its support instances. Specifically, we measure the impact of different instances in the support set for a specific query instance, and build a query-oriented graph with nodes representing different support instances and edges representing their affinities. Notice that the graphs are distinct for different query instances, so this module can be flexibly adapted to different query instances. To construct such query-oriented graph, we first obtain the embeddings for each node as follows:

$$\mathbf{v}_j = \phi_v\left(\mathbf{x}_q \oplus \mathbf{x}_j \oplus (\mathbf{x}_q + \mathbf{x}_j) \oplus (\mathbf{x}_q \odot \mathbf{x}_j)\right), \tag{9}$$

$$\mathbf{V} = (\mathbf{v}_1; \mathbf{v}_2; \ldots; \mathbf{v}_{NK}) \in \mathbb{R}^{NK \times d_h}, \tag{10}$$

where $\mathbf{x}_j$ is the embedding of the j-th support instance in the support set and $\mathbf{x}_q$ is the embedding of a specific query instance (both can be obtained from the neighbor encoder module). ϕ_v is a fully-connected layer that maps the concatenated input to a new embedding space. $\mathbf{V}$ is an $NK \times d_h$ final embedding matrix for different nodes (w.r.t. a specific query instance) and d_h is the embedding size. Besides, $\oplus$ represents the concatenation operator and $\odot$ denotes the element-wise product operation. In this way, we can model the interactions between a specific query instance and different support instances, resulting in a node embedding matrix $\mathbf{V}$ of the query-oriented graph.

Then we apply another fully-connected layer to calculate the similarity matrix of different nodes in the query-oriented graph and normalize it in a row-wise manner:

$$[\mathbf{A}]_{ij} = \phi(\mathbf{v}_i)^T \phi(\mathbf{v}_j), \tag{11}$$

$$\tilde{\mathbf{A}} = \mathrm{softmax}(\mathbf{A}), \tag{12}$$

where $[\mathbf{A}]_{ij}$ denotes the i-th row and the j-th column of matrix $\mathbf{A}$ and the softmax function is applied on each row of $\mathbf{A}$ to ensure the sum of each row is normalized to one. Similar as [47], we make use of a GCN layer with residual connection to measure the trustworthiness of each support instance for the query instance:

$$\mathbf{u} = \mathrm{sigmoid}\left((\mathbf{V} + \tilde{\mathbf{A}}\mathbf{V}\mathbf{W}_v)\mathbf{W}_u\right) \in \mathbb{R}^{NK}, \tag{13}$$

where $\mathbf{W}_v \in \mathbb{R}^{d_h \times d_h}$ and $\mathbf{W}_u \in \mathbb{R}^{d_h}$ are the parameters to learn. In the above formulation, $\tilde{\mathbf{A}}\mathbf{V}\mathbf{W}_v$ propagates information across different nodes in the query-oriented graph and "**V**+" can be considered as a residual connection [39]. Since the graph is fully-connected, one layer is enough for the information propagation. Through the sigmoid function, the output is a vector of length NK with each element denoting the confidence score of each support instance for a specific query instance. Then we take the maximum value along the rows of the reshaped matrix to generate a confidence weight for each relation w.r.t. a specific query instance:

$$\mathbf{U} = \mathrm{reshape}(\mathbf{u}) \in \mathbb{R}^{N \times K}, \tag{14}$$

$$[\mathbf{w}]_i = \max\left\{[\mathbf{U}]_{i1}, [\mathbf{U}]_{i2}, \ldots, [\mathbf{U}]_{iK}\right\}, \tag{15}$$

where $[\mathbf{w}]_i$ denotes the confidence weight for the i-th relation w.r.t. a specific query instance. As such, we can quantify the trustworthiness of each relation for a specific query instance and naturally mitigate the inevitable errors in KGs.

After that, we further make use of the concept of the energy function in KG completion [2, 35, 41] to generate an energy score for each relation i in the support set:

$$s_i = -E(\mathbf{x}_q, \mathbf{r}_i) = -\sigma(\mathbf{W}_e \mathbf{x}_q)^T \sigma(\mathbf{r}_i), \tag{16}$$

where $E(.)$ denotes the energy function, $\mathbf{x}_q$ and $\mathbf{r}_i$ denote the embedding of the j-th query instance and the i-th relation, respectively. $\mathbf{W}_e$ is a trainable weight matrix and σ denotes an activation function which is specified as tanh in our framework.

Lastly, we multiply these energy scores with the confidence weights as the final class assignment probability for each relation:

$$z_i' = \frac{\exp(s_i)}{\sum_{j=1}^{N} \exp(s_j)} [\mathbf{w}]_i, \quad z_i = \frac{z_i'}{\sum_{j=1}^{N} z_i'}, \tag{17}$$

Afterwards, the classification loss for all the query instances in the meta-training task can be formulated by the cross-entropy loss:

$$\mathcal{L} = -\frac{1}{|Q|} \sum_{j=1}^{|Q|} \sum_{i=1}^{N} y_i^j \ln z_i^j, \tag{18}$$

where y_i^j denotes whether the j-th query instance belongs to the i-th relation of the support set in the ground truth, and is either 0 or 1. z_i^j denotes the class assignment probability such that the j-th query instance is assigned to the i-th relation category in the support set (through Eq. (17)).

Algorithm 1 Detailed learning process of REFORM.

Input: A knowledge graph $\mathcal{G}$ consisting of a number of triples $\{(h, r, t)\} \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ (some triples in $\mathcal{G}$ are erroneous), set of few-shot relations $\mathcal{R}_f$ of the meta-test task, set of relations $\mathcal{R}_f'$ of the meta-training tasks, few-shot KG completion task $\mathcal{T}_{test} = \{\mathcal{S}, \mathcal{Q}\}$, T, N, and K.

Output: Predicted relations of query instances in the query set $\mathcal{Q}$.

// `Meta-training phase`

- $i \leftarrow 0$
- **while** $i < T$ **do**
  - Sample a meta-training task $\mathcal{T}_i = \{\mathcal{S}_i, \mathcal{Q}_i\}$ from $\mathcal{R}_{f'}$;
  - Compute representations for instances in $\mathcal{S}_i$ and $\mathcal{Q}_i$ as well as relations in $\mathcal{S}_i$;
  - Generate confidence weights for each relation in $\mathcal{S}_i$ with respect to each query instance in $\mathcal{Q}_i$;
  - Predict relations for query instances in $\mathcal{Q}_i$;
  - Update the model parameters with the meta-training loss of $\mathcal{T}_i$ according to Eq. (18) by one gradient descent step;
  - $i \leftarrow i + 1$
- **end while**

// `Meta-test phase`

- Compute representations for instances in $\mathcal{S}$ and $\mathcal{Q}$ as well as relations in $\mathcal{S}$;
- Generate confidence weights for each relation in $\mathcal{S}$ with respect to each query instance in $\mathcal{Q}$;
- Predict relations for query instances in $\mathcal{Q}$;

3.4 Training Process

The overall training process of our proposed framework REFORM is in Algorithm 1. Firstly we sample T meta-training tasks under the N-way K-shot few-shot learning framework with relations from the relation set $\mathcal{R}_{f'}$. Specifically, for each meta-training task, N relations are sampled from $\mathcal{R}_{f'}$ and meanwhile K instances (e.g., pairs of entities) are sampled from each of the N relations, resulting in a support set. Then another different M instances are also sampled from each of the N relations to form a query set. The support set and the query set together build up a meta-training task. Then we perform predictions on the query set of the meta-training task and finally compute the cross-entropy loss according to Eq. (18). We apply gradient descent methods to update the model parameters by one gradient descent step according to the loss on each meta-training task [8]. And after the training of a total number of T meta-training tasks, we can obtain a fully-trained meta-learning model for the meta-test task. Finally, we apply the learned model on the meta-test task to predict the relations for query instances in the query set $\mathcal{Q}$ of the meta-test task.

4 EXPERIMENTS

In this section, we evaluate the proposed framework on a number of KG datasets which are widely used for KG completion tasks and show that our framework achieves superior performance on the studied problem of error-aware few-shot KG completion. We further demonstrate how different modules of REFORM contribute to the KG completion performance. We first introduce the datasets and experimental settings before presenting details of the experiments[2].

4.1 Datasets

We perform experiments on three KG datasets which are widely used for the KG completion task [41, 43, 45]. The first two datasets are `NELL` and `Wiki`, which are constructed and used by GMatching [43]. The `NELL` dataset is based on the NELL [20], a system that continuously collects information from websites and constructs knowledge from it. The `Wiki` dataset is based on the Wikidata [38], a much larger dataset that has more entities and relations than many other datasets. To further demonstrate the effectiveness of our proposed framework, we also make use of another widely used dataset for the KG completion, the `FB15K-237` dataset [11]. The total number of few-shot relations in these three datasets `NELL`, `Wiki`, and `FB15K-237` are 67, 183, and 119, respectively. We follow the conventional settings of few-shot KG completion [4, 43, 45] to split the training/validation/test relations as 40/5/22, 156/16/11, and 75/11/33 for `NELL`, `Wiki`, and `FB15K-237`, respectively. The detailed statistics of these three datasets are shown in Table 1.

Table 1: Detailed statistics of the used datasets. # Ents. denotes the number of entities, # Rels. denotes the number of all relations, # Triples denotes the number of triples, and # Few-Rels. denotes the number of few-shot relations.

Dataset	# Ents.	# Rels.	# Triples	# Few-Rels.
`NELL`	68,545	358	181,109	67
`Wiki`	4,838,244	822	5,859,240	183
`FB15K-237`	14,541	237	281,624	119

4.2 Experimental Settings

To verify the effectiveness of the proposed framework REFORM, we compare it with the following baseline methods:

- **Prototypical Network** [32]: Prototypical Network encodes the prototype for each class of the support set and matches the query embedding with prototypes.
- **Relation Network** [34]: Relation Network utilizes neural networks to measure the similarity between a query instance and support instances.
- **GMatching** [43]: GMatching encodes entities w.r.t. different neighbors and matches queries with support instance embeddings via a recurrent matching processor [37].
- **FSRL** [45]: FSRL encodes entities with attentions, aggregates instances in a single relation via a recurrent autoencoder aggregation network [10], and matches queries with support instances via a recurrent matching network [37].
- **FAAN** [31]: FAAN encodes instances with attentions for different relations, utilizes transformer encoders within each entity pair, and matches queries with support instances via an attention-based matching processor.

Prototypical Network [32] and Relation Network [34] are conventional few-shot learning methods for i.i.d. data such as image data. To make them applicable to our few-shot KG completion problem, we use concatenated pretrained entity embeddings as the input instance embeddings and then follow the process in the corresponding papers to conduct few-shot learning. GMatching [43], FSRL [45], and FAAN [31] are recently proposed methods for the few-shot KG completion task, but they formulate the problem as a K-shot entity ranking problem for each few-shot relation and cannot be directly compared with our proposed framework. As mentioned previously, we formulate the problem as a relation classification problem which is closer to real-world scenarios where the queries are mainly about missing relations. To make a fair comparison with these methods, we adapt them to the N-way K-shot learning scenario. Specifically, given a head entity, these models make predictions via ranking tail entities by generating scores for them. In the N-way K-shot setting we use their models to produce NK these scores as the classification scores to make predictions. Notice that here we do not compare our model with traditional KG embedding methods such as TransE [2], because in the previous work of GMatching [43] and FSRL [45], the experimental results have already shown that traditional KG embedding methods do not perform well in the few-shot scenario and are outperformed by their models. Therefore, we compare our model with traditional few-shot learning methods which are more suitable for our problem setting.

All the compared models and our proposed REFORM framework are implemented based on PyTorch [24]. Since we formulate the few-shot KG completion as a classification problem, we adopt the *classification accuracy* as the final evaluation metric.

In our experiments, we follow the settings of [31, 43, 45] to split the data into two parts: background KG and few-shot triples. The background KG contains all other triples except for triples belonging to few-shot relations and is used to generate the pretrained TransE [2] embeddings for each entity and relations that are not few-shot. Notice that for each dataset, we make sure all entities in few-shot relations have already appeared in the background KG so that a pretrained embedding is available during training[3]. Then as mentioned in the problem definition part, in the N-way K-shot learning paradigm, N relations as well as K instances for each of them are sampled to form a meta-training task, where N equals to the number of relations in the support set of the meta-test task. Then a number of M query instances are sampled for each meta-training task. Specifically, N in our general experiments equals to the number of relations in the meta-test task. Additionally, we specify $K = 5$ and $M = 5$, and a more detailed analysis of the impact of these two hyperparameters can be found in Section 4.5.

Since we are in short of reliable ground truth on whether a triple is erroneous or not, we randomly inject different levels of errors into the background KG as well as few-shot triples. Specifically, when a background triple or a few-shot triple is selected, we randomly perturb its head entity or the tail entity to another entity to generate

[2]Code and data are available at https://github.com/SongW-SW/REFORM

[3]If random embeddings are used, this constraint could be removed

an incorrect triple. During the perturbation process, we also ensure the generated triple does not belong to the original KG.

For the pretrained embeddings, we adopt the TransE [2] embedding generated by the OpenKE [11] toolkit, where the embedding size d_p is 100 for NELL and FB15K-237 and 50 for Wiki. Then the entity and relation embeddings are trained on the background KG and used as pretrained embeddings for all models in experiments. Notice that the embeddings are trained on the polluted background KG to simulate the real-world scenario, which means the errors will also impact the quality of pretrained embeddings, further bringing challenges to the error-aware few-shot KG completion.

For the hyperparameter settings of our proposed framework, the neighbor encoder adopts a maximum of 50 neighbors for each entity, the transformer encoder includes 3 layers and 4 heads and the dimension of embeddings in the error mitigation module d_h is 100 for NELL and FB15K-237 and 50 for Wiki. For the model optimization, we adopt Adam [16] with the learning rate of 0.001 and a dropout rate of 0.5, and T is set to 5000 for all models.

4.3 Overall Evaluation Results

We first show the performance of error-aware few-shot KG completion by different methods in Table 2. Specifically, to better demonstrate the robustness of our model against errors, we show the results by varying the error rate ranging from 0 to 20%. From the table we can make the following observations:

- Our proposed framework REFORM achieves the best few-shot KG completion performance than any other baselines in all datasets with different error rates, which validates the effectiveness of our proposed framework in completing few-shot KG completion in the presence of errors.
- Conventional few-shot models such as Relation Network [32] and Prototypical Network [34] are not explicitly designed for the KG completion task, and their performance is inferior to the recent few-shot KG completion methods such as GMatching [43], FSRL [45] and FAAN [31].
- When the error rate is increased from 0 to 20%, our proposed framework REFORM has the least performance degradation compared with all other baseline methods. The main reason could be attributed to the fact that our proposed framework explicitly considers the adverse impact of errors in the embedding learning phase by neighbor encoder module and prediction phase by error mitigation module.
- The improvements of few-shot KG completion models (i.e., REFORM, GMatching) over conventional few-shot baselines are relatively higher on the Wiki dataset. The reason is that this dataset contains a larger number of triples than the other two datasets, and few-shot KG completion methods can better exploit the interactions among entities and relations at the learning phase.

In this part, we validate the importance of three essential modules of REFORM by conducting the ablation study on NELL and FB15K-237 (we have similar observations on Wiki). Firstly, we remove the neighbor encoder module and directly use the pretrained embeddings of entities, and we refer this variant as *REFORM w/o neighbor encoder*. The second variant is to remove the cross-relation aggregation module such that we do not consider the correlations between different relations in the support set, and we refer this variant as *REFORM w/o cross-relation aggregation*. The final one is to remove the error mitigation module, which means the confidence weights are the same for all relations, and we refer this variant as *REFORM w/o error mitigation*. The ablation study results are shown in Figure 2 and Figure 3. From the result we can see that all three modules play crucial roles in our proposed framework. Specifically, the removal of the neighbor encoder causes a great decrease in the few-shot KG completion performance. Also, the incorporation of the cross-relation aggregation module brings a decent performance improvement for REFORM. More importantly, when the error mitigation module is not used, the KG completion performance drops quickly when the error rate increases, demonstrating the significant effect of this module in the presence of errors.

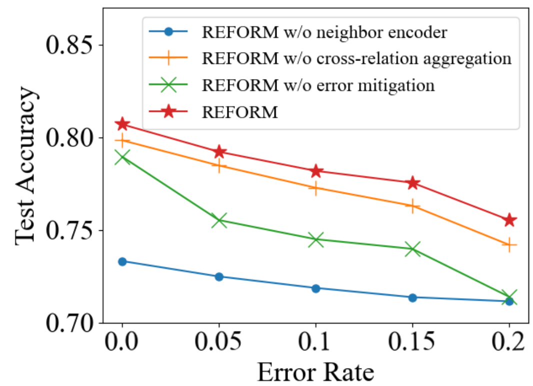

Figure 2: Ablation study of REFORM on NELL

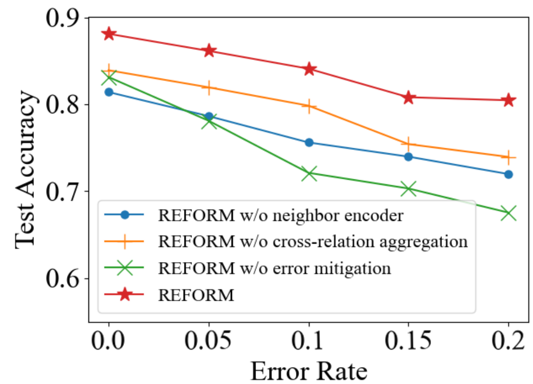

Figure 3: Ablation study of REFORM on FB15K-237

4.4 Effects of K and M in REFORM

In this subsection, we conduct experiments to show how the shot size K (i.e., the number of support instances per relation) and the number of query instances M in each meta-training task affect the performance of REFORM. Figure 4 reports the results of REFORM when varying the value of K and M on the NELL dataset with an error rate of 0. Specifically, M is set to 5 when we vary the value of K, and similarly, K is set to 5 when the value of M is changed. From

Table 2: The overall error-aware few-shot KG completion results (accuracy in %) of various models under different error rates.

Dataset	NELL					Wiki					FB15K-237				
Error Rate	0	5%	10%	15%	20%	0	5%	10%	15%	20%	0	5%	10%	15%	20%
Relation Network	53.89	53.76	46.74	45.79	40.18	47.54	43.03	40.68	39.98	37.73	62.21	61.94	55.31	52.97	52.41
Proto Network	69.59	68.07	65.06	64.29	63.23	61.26	58.13	57.28	54.80	53.44	84.06	83.20	81.30	78.54	78.03
GMatching	77.50	76.52	75.22	73.43	71.94	71.07	68.63	66.13	65.50	63.35	84.75	84.40	81.76	79.05	78.67
FSRL	75.78	72.63	72.55	71.70	70.31	59.57	47.92	47.34	43.86	39.47	83.27	79.72	78.08	73.86	72.85
FAAN	77.10	76.33	74.80	74.11	70.83	70.19	64.87	63.61	62.57	59.34	86.32	85.19	83.31	78.63	77.23
REFORM (Ours)	**80.70**	**79.22**	**78.19**	**77.55**	**75.54**	**75.03**	**71.90**	**70.84**	**70.19**	**68.50**	**88.10**	**86.14**	**84.09**	**80.81**	**80.46**

Table 3: The few-shot KG completion results for each few-shot relation of different methods in the test set of NELL dataset.

Relations	REFORM	GMatching	FAAN	FSRL	Relations	REFORM	GMatching	FAAN	FSRL
SportsGameSport	**1.0000**	**1.0000**	**1.0000**	**1.0000**	CountryStates	**0.8476**	0.6341	0.8293	0.8293
CarDealerCity	**1.0000**	0.9711	**1.0000**	**1.0000**	StateOfHeadquarter	**0.9826**	0.9565	**0.9826**	0.9391
AthleteInjured	**1.0000**	0.8438	0.9375	0.7969	FoodCauseDisease	**1.0000**	0.9615	0.9808	0.9423
PoliticalLocation	**0.3623**	0.2754	0.2101	0.1232	OrganizationAs	**0.3929**	0.3036	0.3036	0.2679
PoliticianEndorses	**0.9816**	0.9501	0.9790	0.9711	AnimalEatVegetable	0.9595	0.9364	**1.0000**	0.7746
AnimalInvertebrate	**0.2366**	0.1488	0.0585	0.0951	FatherOfPerson	**0.9029**	**0.9029**	0.7767	0.8252
SportSchoolCountry	**1.0000**	**1.0000**	**1.0000**	**1.0000**	CountryCapital	**0.9375**	0.9135	0.8894	0.6154
AgriculturalProduct	**0.9852**	0.8519	**0.9852**	0.9704	PersonMoveToState	0.9909	0.9545	**0.9955**	0.9727
CarDealerCountry	**1.0000**	0.9890	0.8681	0.9670	PersonKnownAs	**0.2329**	0.1233	0.2192	0.1096
TeamCoach	**0.9970**	0.9940	0.9911	0.9821	OfficeInCountry	0.0000	0.0719	0.0144	**0.2590**
ProducedBy	**0.9856**	0.9519	0.9760	**0.9856**	AnimalEatInsect	0.9700	0.9528	**0.9957**	0.8112

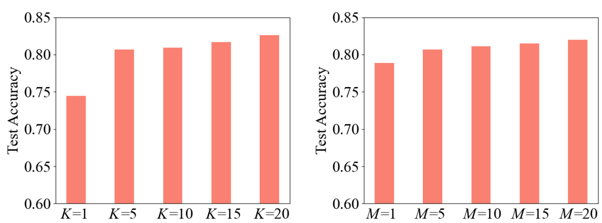

Figure 4: Results with different K (left) and M (right) on NELL

the figure we can find that the few-shot KG completion results increase when K increases. The reason is that a larger support set helps produce better embeddings for few-shot relations. Also, the error mitigation module becomes more powerful with more support instances and thus benefits the final results. Also, involving more query instances during training (i.e., increasing the value of M) slightly increases the performance as a larger training set helps alleviate the over-fitting problem.

4.5 Results on Different Few-shot Relations

Aside from the overall results from all few-shot relations in Table 2, we further conduct experiments to evaluate the effectiveness of REFORM for each relation in the NELL dataset with an error rate of 15%. Table 3 shows the results of our framework and other few-shot KG completion baselines. The best results are highlighted in bold. From the results we can find that each model performs distinctly in different relations. The reason may be that some relations contain entities that appear much more times in other relations, which reduces the impact of errors due to their interactions with other relations. Even so, our proposed framework REFORM achieves the best performance in $18/22 \approx 81.82\%$ relations while the runner-up FAAN only achieves the best performance in half of the relations.

5 RELATED WORK

We review related works from three aspects: (1) relational learning for knowledge graphs; (2) error detection in knowledge graphs; and (3) few-shot learning in knowledge graphs.

5.1 Relational Learning for Knowledge Graphs

Recently, various kinds of embeddings models have been proposed to represent entities and relations in knowledge graphs as continuous vectors for knowledge graph completion tasks. The general idea is to learn embeddings for entities and relations and then compute ranking scores for prediction. Among them, RESCAL [23] is one of the classic models, which models the relationships using tensor operations. TransE [2] models relationships as translating operation on embeddings of entities in the vector space. NTN [33] combines linear transformations and multiple bilinear forms of entities for expressiveness while having a large number of parameters. Besides, ComplEx [35] is also proposed as an advanced model. More recently, deep neural networks have also been applied on these embedding models. For example, ConvE [5] utilizes convolutional neural networks for further improvements.

Although deep neural network based embedding models have shown strong performance in learning representations for entities and relations in knowledge graph completion tasks, they usually assume that there are adequate training instances for all entities and relations, which is far from the real-world scenario as described previously. Besides, they do not consider about the situation that a certain number of inevitable errors exist in real-world knowledge graphs, which may jeopardize the performance of these approaches potentially. To better fit in the real-world scenario and take possibly existing errors into consideration, our model proposes to conduct knowledge graph completion tasks with very limited number of training instances while maintaining considerable robustness against errors, which is important since the impacts of errors will possibly be amplified due to scarce training instances.

5.2 Error Detection in Knowledge Graphs

The error detection task in knowledge graphs, which is also referred to as noise detection, is to detect the erroneous information such as incorrect relations or incorrect head/tail entities. The necessity of this task lies on the fact that errors are inevitably incorporated along with the process of constructing knowledge graphs (regardless of manually or automatically constructed) [18]. For example, YAGO 4 [26]—a large semantic knowledge base, is derived from Wikidata with its accuracy manually evaluated. The NELL [20] system continuously extracts facts from webs to construct its knowledge base while automatically assigning confidence values to candidate triples using heuristics and constantly updating them with human supervision. These examples exhibit a situation that manual detection is extremely labor-intensive and thus necessitate the research on automatic error detection [22] in KGs.

Some recent methods heavily rely on external information to perform error detection and the error detection performance is largely dependent on the quality of external information. For example, Knowledge Vault [7] estimates a probability score of reliability to determine the quality of a triple via several prior models fitted with existing KGs and web contents. The similar concept of judgments for each triple is also applied in CKRL [42] and NKRL [30], in which the values are called confidences. The former model generates confidence values for triples via internal structure information and utilizes them in representation learning to produce robust representations, while the latter improves it in the aspect of sample selection. The authors in [15] propose to learn the confidence in a supervised manner by measuring correctness at three levels, which are entity level, relationship level, and KG global level. Different from these methods, our model proposes to reduce the influence brought by potentially existing errors in knowledge graphs in the few-shot scenarios, where errors are more likely to have impacts on performance since the number of training instances is very limited.

5.3 Few-shot Learning in Knowledge Graphs

Few-shot learning aims to obtain considerable classification performance while using very few training samples for each class. In general, few-shot learning methods can be divided into two categories: metric-based models and meta-optimizer-based models. The former kind targets at learning an effective metric as well as a suitable matching function to measure the distance between classes. As a classic example, Matching Networks [37] output predictions via the similarity between the query sample and each support sample. And Prototypical Networks [32] propose to generate a prototype representation for each class and measure its distance to the query sample. The latter kind aims to optimize the model parameters via gradients on few-shot samples to make the model quickly generalize to new concepts. Among them, MAML [8] updates parameters with few gradients and conducts fast learning on new data while LSTM-based meta-learner [29] learns the step size during training to optimize the model parameters.

In the field of knowledge graphs, some models are proposed recently for few-shot relational learning. GMatching [43], FSRL [45] and FAAN [31] propose to handle the challenge in few-shot KG completion with GMatching focusing on one-shot learning and the other two models dedicating to deeper model structures with attention mechanism. Both methods make use of neighbor information of entities and output predictions for each few-shot relation. Although attention mechanism is also utilized in our model, the purposes are different from previous works. In the cross-relation aggregation, the attention mechanism is applied within the transformer block to fully take advantage of correlations between relations. And in the error mitigation module, the confidence weights can be considered as attention weights for reducing the impact of errors. MetaR [4] utilizes meta-learning on this task by extracting relation-specific meta information and uses it for few-shot relational predictions. ZS-GAN [27] proposes to make use of generative adversarial networks [28] to conduct zero-shot relational learning via generating relation embeddings from texts. It should be noted that previous few-shot learning works in knowledge graphs only aim at predicting entities under a given relation while ignoring potential errors during training. To the best of our knowledge, we are the first to study error-aware few-shot knowledge graph completion and focus on the relation prediction which is closer to the real-world scenario.

6 CONCLUSION

In this paper, we study a novel problem of error-aware few-shot KG completion due to the widely existing errors and the long tail distribution of relations in KGs and present a novel framework REFORM. We formulate the problem under the few-shot learning framework and propose to accumulate meta-knowledge across different meta-tasks to effectively conduct error-aware few-shot KG completion. To address the associated challenges resulting from insufficient training samples and inevitable errors, REFORM utilizes three essential modules: neighbor encoder, cross-relation aggregation, and error mitigation to perform effective encoding with erroneous neighbors, capture correlations between relations, and reduce the impacts of errors during predictions, respectively. Extensive experimental results on three widely used KG datasets demonstrate that REFORM outperforms many state-of-the-art baseline methods. Besides, the ablation study also verifies the effectiveness of each module in the proposed framework. Toward this newly presented task, there are still a lot of challenges needed to be addressed in the future. Future work may consider alleviating the effect of errors in the query set in meta-training tasks or utilizing external information to further improve the model robustness against errors.

7 ACKNOWLEDGMENTS

Jundong Li is partially supported by the National Science Foundation (NSF) under grants #2006844.

REFERENCES

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: a nucleus for a web of open data. In *The Semantic Web*. Springer, 722–735.

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*. 2787–2795.

[3] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the 2010 AAAI Conference on Artificial Intelligence*. 1306–1313.

[4] Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. 2019. Meta relational learning for few-shot link prediction in knowledge graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 4216–4225.

[5] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[6] Kaize Ding, Jianling Wang, Jundong Li, Kai Shu, Chenghao Liu, and Huan Liu. 2020. Graph prototypical networks for few-shot learning on attributed networks. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. 295–304.

[7] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 601–610.

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*. 1126–1135.

[9] Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *Proceedings of the 2019 AAAI Conference on Artificial Intelligence*. 6407–6414.

[10] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*. 1024–1034.

[11] Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. Openke: an open toolkit for knowledge embedding. In *Proceedings of the 2018 conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 139–144.

[12] Xu Han, Zhiyuan Liu, and Maosong Sun. 2018. Neural knowledge acquisition via mutual attention between knowledge graph and text. In *Proceedings of the 2018 AAAI Conference on Artificial Intelligence*. 4832–4839.

[13] Stefan Heindorf, Martin Potthast, Benno Stein, and Gregor Engels. 2016. Vandalism detection in wikidata. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 327–336.

[14] Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*. 105–113.

[15] Shengbin Jia, Yang Xiang, Xiaojun Chen, and Kun Wang. 2019. Triple trustworthiness measurement for knowledge graph. In *Proceedings of the 2019 Web Conference*. 2865–2871.

[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[17] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 2017 International Conference on Learning Representations*.

[18] Michel Manago and Yves Kodratoff. 1987. Noise and knowledge acquisition. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence*. 348–354.

[19] Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 777–782.

[20] Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhavana Dalvi, Matt Gardner, Bryan Kisiel, et al. 2018. Never-ending learning. *Commun. ACM* 61, 5 (2018), 103–115.

[21] Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. Learning attention-based embeddings for relation prediction in knowledge graphs. In *Proceedings of the 57th Conference of the Association for Computational Linguistics5*. 4710–4723.

[22] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A review of relational machine learning for knowledge graphs. *Proc. IEEE* 104, 1 (2015), 11–33.

[23] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning*. 809–816.

[24] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[25] Heiko Paulheim. 2017. Knowledge graph refinement: a survey of approaches and evaluation methods. *Semantic web* 8, 3 (2017), 489–508.

[26] Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. 2020. Yago 4: a reason-able knowledge base. *The Semantic Web* (2020).

[27] Pengda Qin, Xin Wang, Wenhu Chen, Chunyun Zhang, Weiran Xu, and William Yang Wang. 2020. Generative adversarial zero-shot relational learning for knowledge graphs. (2020), 8673–8680.

[28] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proceedings of the 4th International Conference on Learning Representations*.

[29] Sachin Ravi and Hugo Larochelle. 2016. Optimization as a model for few-shot learning. In *Proceedings of the 5th International Conference on Learning Representations*.

[30] Yingchun Shan, Chenyang Bu, Xiaojian Liu, Shengwei Ji, and Lei Li. 2018. Confidence-aware negative sampling method for noisy knowledge graph embedding. In *Proceedings of the 2018 IEEE International Conference on Big Knowledge*. 33–40.

[31] Jiawei Sheng, Shu Guo, Zhenyu Chen, Juwei Yue, Lihong Wang, Tingwen Liu, and Hongbo Xu. 2020. Adaptive attentional network for few-shot knowledge graph completion. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. 1681–1691.

[32] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*. 4077–4087.

[33] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*. 926–934.

[34] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: relation network for few-shot learning. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*. 1199–1208.

[35] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 2016 International Conference on Machine Learning*. 2071–2080.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[37] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*. 3630–3638.

[38] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

[39] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-local neural networks. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*. 7794–7803.

[40] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 950–958.

[41] Han Xiao, Minlie Huang, Yu Hao, and Xiaoyan Zhu. 2016. Transg: a generative mixture model for knowledge graph embedding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. 2316–2325.

[42] Ruobing Xie, Zhiyuan Liu, Fen Lin, and Leyu Lin. 2017. Does william shakespeare really write hamlet? knowledge representation learning with confidence. In *Proceedings of the 32th AAAI Conference on Artificial Intelligence*. 4954–4961.

[43] Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2018. One-shot relational learning for knowledge graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 1980–1990.

[44] Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018. Diverse few-shot text classification with multiple metrics. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1206—-1215.

[45] Chuxu Zhang, Huaxiu Yao, Chao Huang, Meng Jiang, Zhenhui Li, and Nitesh V Chawla. 2019. Few-shot knowledge graph completion. In *Proceedings of the 2020 AAAI Conference on Artificial Intelligence*. 3041–3048.

[46] Fuxiang Zhang, Xin Wang, Zhao Li, and Jianxin Li. 2020. TransRHS: a representation learning method for knowledge graphs with relation hierarchical structure. In *Proceedings of the 2020 International Joint Conference on Artificial Intelligence*. 2987–2993.

[47] Zhizheng Zhang, Cuiling Lan, Wenjun Zeng, Zhibo Chen, and Shih-Fu Chang. 2020. Uncertainty-aware few-shot image classification. (2020).