

Few-shot Low-resource Knowledge Graph Completion with Reinforced Task Generation

Shichao Pei^{1,*}, Qiannan Zhang^{2,*}, Xiangliang Zhang¹

¹University of Notre Dame

²King Abdullah University of Science and Technology
{spei2, xzhang33}@nd.edu, qiannan.zhang@kaust.edu.sa

Abstract

Despite becoming a prevailing paradigm for organizing knowledge, most knowledge graphs (KGs) suffer from the low-resource issue due to the deficiency of data sources. The enrichment of KGs by automatic knowledge graph completion is impeded by the intrinsic long-tail property of KGs. In spite of their prosperity, existing few-shot learning-based models have difficulty alleviating the impact of the long-tail issue on low-resource KGs because of the lack of training tasks. To tackle the challenging long-tail issue on low-resource KG completion, in this paper, we propose a novel few-shot low-resource knowledge graph completion framework, which is composed of three components, i.e., few-shot learner, task generator, and task selector. The key idea is to generate and then select the beneficial few-shot tasks that complement the current tasks and enable the optimization of the few-shot learner using the selected few-shot tasks. Extensive experiments conducted on several real-world knowledge graphs validate the effectiveness of our proposed method.

1 Introduction

Recent years have witnessed a rapidly increasing amount of research attention and industry demand on knowledge graphs (KGs), which organize knowledge in the form of triples and have been playing a crucial role in many knowledge-intensive applications, such as question answering (Zhang et al., 2018), recommendation systems (Guo et al., 2020), and dialogue systems (Yang et al., 2020).

Despite the fact that several large-scale KGs, such as NELL (Mitchell et al., 2018) and Freebase (Bollacker et al., 2008), have been well-developed for organizing general knowledge, most of the existing KGs are built for domain-specific knowledge using domain-specific data sources. These KGs inevitably suffer from the **low-resource issue**, i.e.,

the incompleteness in terms of entities, relations, and triples that exist in the real world but are absent in KGs, caused by the deficiency of data sources, especially those in non-global languages. For example, the number of articles in Norwegian Wikipedia is 10 times less than that in English Wikipedia¹. A KG built using Norwegian Wikipedia would have a severer incompleteness issue compared with the KG extracted from English Wikipedia.

Although automatic KG completion (KGC) (Zhang et al., 2020c; Kazemi and Poole, 2018; Bordes et al., 2013; Tang et al., 2022) has empowered the discovery of more triples to enrich KGs, it is far from perfect to enrich low-resource KGs due to the difficulty in the discovery that stems from the relations associated with missing triples and varies with the change of relation frequency. Since most KGs generally follow **long-tail distribution** (Xiong et al., 2018; Zhang et al., 2020a; Nguyen et al., 2018), where a large fraction of relations have only a few triples, the existence of rare relations on low-resource KGs leads to dramatic performance degradation on the discovery of missing triples and impede the development of an effective completion model. A line of efforts (Xiong et al., 2018; Zhang et al., 2020a; Chen et al., 2019; Lv et al., 2019) attempt to improve the capability of inference on rare relations by formulating the KG completion problem into a few-shot learning framework and exploiting inference models for frequent relations to facilitate inference on rare relations. Such kind of methods require abundant training tasks to ease the effect of memorization and improve the generalization (Rajendran et al., 2020). However, low-resource KGs do not always have adequate training tasks for mimicking the few-shot learning scenarios of rare relations. For example, the Greek KG (Chen et al., 2020) used in experiments only contains 21 training tasks, as many relations are unknown or cannot offer triples for

*Equal contribution.

¹https://en.wikipedia.org/wiki/List_of_Wikipedias

training task construction. Therefore, the few-shot KG completion models tend to overfit on the scarce tasks and engender unsatisfactory results.

Motivated by the above-discussed limitation on low-resource KGs, we propose to **generate new few-shot tasks to augment current tasks for easing the impact of memorization and improving the generalization**. Although straightforward, the materialization of this idea is non-trivial due to two crucial challenges. The first challenge (**C1**) is to design an effective strategy to generate new few-shot tasks. A few-shot task is formed as a set of pairs corresponding to the same relation, each of which includes two entities, i.e., head entity and tail entity. On the one hand, the discrete nature of tasks to generate hinders the harness of prevailing generative models. On the other hand, the generation of few-shot tasks with novel relations is arduous due to the lack of related data samples. If possible to generate, the second challenge (**C2**) is to ensure the authenticity of generated few-shot tasks. The generated tasks ineluctably consist of noisy tasks which cannot represent the designated relation and would be detrimental to few-shot learning. Thus, how to select beneficial tasks is an essential problem.

In this paper, we propose a novel few-shot low-resource KG completion framework with reinforced task generation, named **FLow-KGC**, to promote the KG completion on rare relations in low-resource KGs. Specifically, we formulate KG completion into a few-shot learning framework and train a simple yet effective *few-shot learner* to learn representations for rare relations with a small support set. The learned representation can be further used for inference. To address challenge C1, we represent few-shot tasks in a latent continuous space and then build a *task generator* in this latent space, rather than generating original discrete structure samples. Then the generated tasks can be utilized to update the few-shot learner. To select beneficial synthetic few-shot tasks and tackle challenge (C2), we design an adaptive *task selector*, which makes decisions on keeping or discarding generated tasks and receives feedback from the few-shot learner and conducts optimization using reinforcement learning. The task generator and task selector collaboratively complete the process of task generation. These three components, i.e., few-shot learner, task generator, and task selector, constitute our proposed method **FLow-KGC** and can be updated in an alternative optimization way.

Overall, our contributions in this work include:

- (1) We study the crucial few-shot KG completion on low-resource KGs and propose a novel model called **FLow-KGC** to mitigate the impact of the long-tail problem on completion tasks.
- (2) We design a task generator to create synthetic few-shot tasks and a task selector to achieve adaptive beneficial task selection, in order to improve the generalization of the few-shot learner.
- (3) We perform extensive experiments on several real-world low-resource KGs. The experimental results show the superior performance of FLow-KGC over the state-of-the-art with significant improvement in few-shot low-resource KG completion.

2 Related Work

2.1 Knowledge Graph Completion

Early efforts address KGC by designing algorithms based on rule learning (Galárraga et al., 2015) to discover logical rules from KGs for facilitating inductive link prediction. Nowadays, the prevailing learning paradigm for KG completion tasks (Zhang et al., 2020c) is to learn the distributed representation of entities and relations in KGs. These methods roughly fall into three categories (Bonner et al., 2021): 1) Tensor decomposition methods, such as Simple (Kazemi and Poole, 2018), RESCAL (Nickel et al., 2012), and ComplEx (Trouillon et al., 2017). 2) Geometric methods, such as TransE (Bordes et al., 2013), RotatE (Sun et al., 2018b), and CrossE (Zhang et al., 2019). 3) Deep learning methods (Nguyen et al., 2018; Dettmers et al., 2018; Vashishth et al., 2020). However, these methods fail to handle the long-tail problem of KG completion and suffer from performance degradation of prediction on rare relations. A recent approach (Zhang et al., 2020b) attempts to alleviate this issue, but it requires a relatively large set of triples for rare relations and is incapable of handling the few-shot scenario. Recently, a few works (Chen et al., 2020; Zhou et al., 2021) design strategies to assist the KG completion by leveraging complementary knowledge from other related KGs in different languages (Chen et al., 2020; Conneau et al., 2020; Zhou et al., 2021). Although effective, they only rely on cross-lingual links (Chen et al., 2017; Sun et al., 2018a; Pei et al., 2019a; Cao et al., 2019; Pei et al., 2019b, 2020) and cannot handle new relations or rare relations with few triples.

2.2 Few-shot Knowledge Graph Completion

With the goal to overcome the shortcoming of canonical KG completion methods (Kazemi and Poole, 2018; Bordes et al., 2013; Nguyen et al., 2018) on inferring the missing triples associated with rare relations, several few-shot KG completion (FSKGC) algorithms have been developed for improving completion performance on rare relations. Recent attempts (Xiong et al., 2018; Zhang et al., 2020a; Chen et al., 2019; Lv et al., 2019) on few-shot KG completion formulate the problem into a few-shot learning framework and propose metric-based approaches (Xiong et al., 2018; Zhang et al., 2020a; Sheng et al., 2020; Niu et al., 2021) and meta-learning-based approaches (Chen et al., 2019; Lv et al., 2019). Yet these few-shot KG completion models require plenty of training tasks to train a few-shot learner and cannot generalize well to rare relations in low-resource KGs due to the deficiency of relations and corresponding tasks.

3 Problem Formulation

A knowledge graph \mathcal{G} can be denoted as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{TP})$, where \mathcal{E} refers to the set of entities and \mathcal{R} denotes the set of relations, and \mathcal{TP} is represented as a set of triples $\{(h, r, t)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, each of which includes a head entity h , a relation r , and a tail entity t . The KG completion problem is to infer the most plausible missing triples from the candidate set $\{(h, r, t) | t \in \mathcal{E} \wedge (h, r, t) \notin \mathcal{TP}\}$ for each incomplete triple $(h, r, ?)$ (or inferring from $\{(h, r, t) | h \in \mathcal{E} \wedge (h, r, t) \notin \mathcal{TP}\}$ for $(?, r, t)$).

In our few-shot low-resource KGC problem, the majority of relations have a few training triples. Formally, for a rare relation r_i , there are a support set $\mathcal{S}_i = \{(h, t) | (h, r_i, t) \in \mathcal{TP}\}$ including only $|\mathcal{S}_i|$ training triples. The task is to predict the tail entity for a query triple $(h, r_i, ?)$, i.e., ranking all tail entity candidates such that the true tail entity t is ranked higher than other candidates in \mathcal{C}_{h, r_i} , which is defined as $\mathcal{C}_{h, r_i} = \{t_c | t_c \in \mathcal{E} \wedge (h, r_i, t_c) \notin \mathcal{TP}\}$. Following the notations in meta-learning, a few-shot task with $K = |\mathcal{S}_i|$ triples is called K -shot KG completion. Since this K for a rare relation r_i is often small (e.g., less than 10), the learning problem is organized by working on meta-tasks $\mathcal{T}_i = \{\mathcal{S}_i, \mathcal{Q}_i\}$, where a query set $\mathcal{Q}_i = \{(h, t, \mathcal{C}_{h, r_i}) | (h, r_i, t) \in \mathcal{TP}\}$.

In the meta-training stage, there are a set of few-shot tasks $\mathcal{T}_{train} = \{\mathcal{T}_i\}_{i=1}^M$, where each task \mathcal{T}_i represents a relation $r_i \in \mathcal{R}_{train}$. A few-shot learn-

ing model can be optimized using meta-training tasks following the meta-learning principle (Finn et al., 2017). In the meta-testing stage, with a set of testing tasks $\mathcal{T}_{test} = \{\mathcal{T}_j\}_{j=1}^J$, the model should make inferences on new relation $r_j \in \mathcal{R}_{test}$ corresponding to task \mathcal{T}_j . Note that all relations that appear in the meta-testing tasks are unseen during meta-training, i.e., $\mathcal{R}_{train} \cap \mathcal{R}_{test} = \emptyset$.

Low-resource KGs suffer from severer incompleteness issue and consist of fewer relations. Due to the long-tail distribution of these relations, there are only a small number of relations with a high frequency qualified to offer triples for task construction, leading to insufficient \mathcal{T}_{train} . Our proposed solution enriches \mathcal{T}_{train} by generating and selecting beneficial new tasks for virtual relations which are undiscovered and not present in KGs.

4 The Proposed Method

In this section, we introduce the proposed method **FLoW-KGC**, describe each component in detail, and elaborate on the optimization and inference, following the overview shown in Figure 1.

4.1 Few-shot Learner

A few-shot learner is a basic yet essential component in few-shot learning for generalizing a inference model over rare relations with only a few triples. In particular, the few-shot learner should be able to learn accurate representation for rare relations using a small support set, then the learned representation will be applied for subsequent inference. We adopt a simple few-shot learner based on meta-learning (Chen et al., 2019) to learn representation for relations. Specifically, given a meta-training task \mathcal{T}_i with support set \mathcal{S}_i and its corresponding relation $r_i \in \mathcal{R}_{train}$, a few-shot learner $\text{FS}(\cdot)$ aims to generate a representation for r_i by taking entities in associated triples as input. The representation of r_i can be obtained by the following:

$$\mathbf{r}_{\mathcal{T}_i} = \frac{\sum_{k=1}^{|\mathcal{S}_i|} \text{FS}(\mathbf{e}_{h_k} \oplus \mathbf{e}_{t_k})}{|\mathcal{S}_i|}, \quad (1)$$

where (h_k, t_k) denotes the k -th pair in \mathcal{S}_i , \mathbf{e}_{h_k} and \mathbf{e}_{t_k} are embeddings for entity h_k and t_k , and $\mathbf{r}_{\mathcal{T}_i}$ refers to the learned representation of relation r_i . $\mathbf{x} \oplus \mathbf{y}$ denotes the concatenation of the embedding \mathbf{x} and \mathbf{y} . The few-shot learner $\text{FS}(\cdot)$ is implemented by an L -layers fully connected neural network.

After obtaining the representation $\mathbf{r}_{\mathcal{T}_i}$ of relation r_i , we measure if $\mathbf{r}_{\mathcal{T}_i}$ can well represent relation

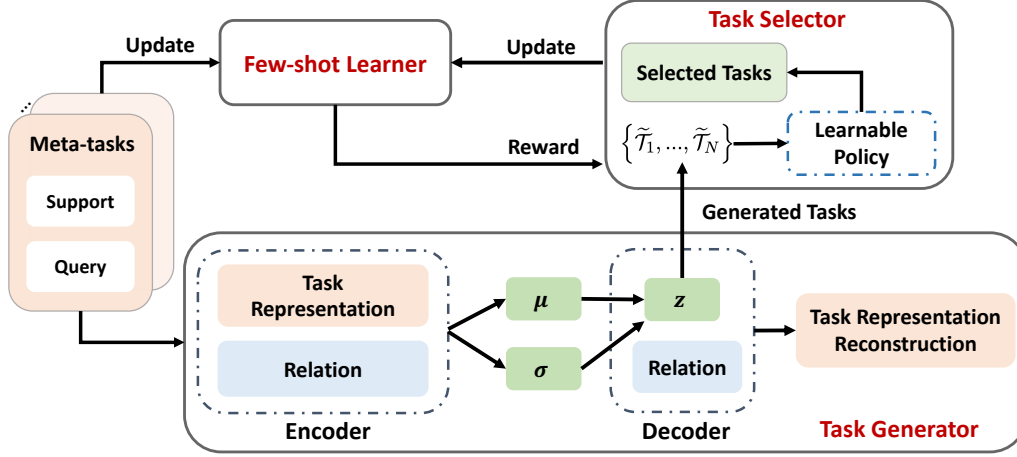


Figure 1: An overview of the FFlow-KGC model. (1) The few-shot learner learns the representation for relations based on given meta-tasks. (2) The task generator estimates the latent distribution of meta-tasks and generates synthetic tasks. (3) The task selector makes decisions on keeping or discarding generated tasks and feeds them to update the few-shot learner. Then feedback from the few-shot learner can guide the optimization of the task selector.

r_i by a score function $f^s(\cdot)$ and define the meta-training loss on support set \mathcal{S}_i as follow:

$$\mathcal{L}_{\mathcal{S}_i} = \sum_{k=1}^{|\mathcal{S}_i|} [\gamma + f^s(h_k, r_i, t_k) - f^s(h_k, r_i, t_n)]_+, \quad (2)$$

where $f^s(\cdot)$ is based on TransE (Bordes et al., 2013) algorithm and measures the plausibility of triple (h_k, r_i, t_k) by $f^s(h_k, r_i, t_k) = \|\mathbf{e}_{h_k} + \mathbf{r}_{\mathcal{T}_i} - \mathbf{e}_{t_k}\|$. $f^s(h_k, r_i, t_n)$ is the score for negative sample (h_k, t_n) sampled from $\{(h_k, t') | t' \in \mathcal{E} \cup (h_k, r_i, t') \notin \mathcal{TP}\}$. And γ is a predefined margin parameter. Then a fast update on $\mathbf{r}_{\mathcal{T}_i}$ can be conducted to obtain a more accurate representation of relations and works as follows:

$$\hat{\mathbf{r}}_{\mathcal{T}_i} = \mathbf{r}_{\mathcal{T}_i} - \beta \nabla_{\mathbf{r}_{\mathcal{T}_i}} \mathcal{L}_{\mathcal{S}_i}, \quad (3)$$

where β is the step size of the gradient updates.

Next, the updated relation representation $\hat{\mathbf{r}}_{\mathcal{T}_i}$ is exploited to measure the plausibility of triples in the query set \mathcal{Q}_i with a score function $f^q(\cdot)$ defined by $f^q(h_z, r_i, t_z) = \|\mathbf{e}_{h_z} + \hat{\mathbf{r}}_{\mathcal{T}_i} - \mathbf{e}_{t_z}\|$, where (h_z, t_z) denotes the z -th pair in \mathcal{Q}_i . Then the loss function for updating few-shot learner FS(\cdot) on the query set \mathcal{Q}_i is defined as follows:

$$\mathcal{L}_{\mathcal{Q}_i} = \sum_{z=1}^{|\mathcal{Q}_i|} [\gamma + f^q(h_z, r_i, t_z) - f^q(h_z, r_i, t_c)]_+, \quad (4)$$

where $t_c \in \mathcal{C}_{h_z, r_i}$ is a tail entity candidate.

Lastly, the few-shot learner is optimized on all

meta-training tasks as follows:

$$\mathcal{L}_{\text{FS}} = \sum_{i=1}^{|\mathcal{T}_{\text{train}}|} \mathcal{L}_{\mathcal{Q}_i}. \quad (5)$$

4.2 Task Generator

The few-shot learner on low-resource KGs suffering from the deficiency of training tasks tends to overfit and memorize the given tasks and lacks generalization ability. We thus target to generate few-shot tasks to complement the meta-training set. In particular, We design a task generator in a task representation space, rather than generating new tasks in the original discrete format.

4.2.1 Task Representation

We build the representation of a task based on the entities involved in the task. Specifically, for a task \mathcal{T}_i with a support set \mathcal{S}_i and a query set \mathcal{Q}_i , we denote the representations of \mathcal{S}_i and \mathcal{Q}_i as $\mathbf{S}_i \in \mathbb{R}^{K \times 2d}$ and $\mathbf{Q}_i \in \mathbb{R}^{Z \times 2d}$, which are obtained by:

$$\begin{aligned} \mathbf{S}_i &= \mathbf{S}\mathbf{H}_i \oplus \mathbf{S}\mathbf{T}_i \\ \mathbf{Q}_i &= \mathbf{Q}\mathbf{H}_i \oplus \mathbf{Q}\mathbf{T}_i, \end{aligned} \quad (6)$$

where $\mathbf{S}\mathbf{H}_i \in \mathbb{R}^{K \times d}$ denotes the representation matrix of head entities in support set \mathcal{S}_i and is obtained by $\mathbf{S}\mathbf{H}_i = \mathbf{e}_{h_1} \oplus \dots \oplus \mathbf{e}_{h_K}$. d is the dimension of embeddings. Similarly, $\mathbf{S}\mathbf{T}_i \in \mathbb{R}^{K \times d}$ refers to the representation matrix of tail entities in support set \mathcal{S}_i and is obtained by $\mathbf{S}\mathbf{T}_i = \mathbf{e}_{t_1} \oplus \dots \oplus \mathbf{e}_{t_K}$. K denotes the size of the support set. The representation of \mathcal{Q}_i can be acquired in the same way. $\mathbf{Q}\mathbf{H}_i$

and \mathbf{QT}_i represent the head entities and tail entities in \mathcal{Q}_i , respectively. Z denotes the size of the query set. With obtained \mathbf{S}_i and \mathbf{Q}_i , a task \mathcal{T}_i can be represented by $\mathbf{T}_i \in \mathbb{R}^{(K+Z) \times 2d}$ as follow:

$$\mathbf{T}_i = \mathbf{S}_i \oplus \mathbf{Q}_i. \quad (7)$$

4.2.2 Conditional Variational Autoencoder

With learned task representations and the aim to generate meta-tasks, we expect to estimate the underlying posterior distribution $p(\cdot|\mathbf{T}, r)$ of meta-tasks with given relations for task generation by sampling. Due to the intractability of this posterior distribution, we employ a conditional variational autoencoder (CVAE) (Sohn et al., 2015) to circumvent the direct posterior estimation and generate meta-tasks with relations as a conditional variable.

Encoder. The encoder takes the representation \mathbf{T} of a task \mathcal{T} with its corresponding relation r as input data and aims to construct a latent distribution $q_\phi(\mathbf{z}|\mathbf{T}, r)$ to describe the distribution of meta tasks related to relation r , which is represented by the mean μ and standard deviation σ , and \mathbf{z} is a sample drawn from the distribution. μ and σ are learned by two separate linear function f_μ and f_σ as follows:

$$\mu = f_\mu(\text{MLP}_{\text{enc}}(\text{Concat}(\mathbf{T}, r))) \quad (8)$$

$$\sigma = f_\sigma(\text{MLP}_{\text{enc}}(\text{Concat}(\mathbf{T}, r))), \quad (9)$$

where $f_\mu(\mathbf{x}) = \mathbf{W}_\mu \mathbf{x} + \mathbf{b}_\mu$ and $f_\sigma(\mathbf{x}) = \mathbf{W}_\sigma \mathbf{x} + \mathbf{b}_\sigma$, $\text{MLP}_{\text{enc}}(\cdot)$ is an L -layer fully connected neural network, $\text{Concat}(\cdot)$ is a concatenation operator, which first squeezes \mathbf{T} to a vector $\mathbf{T}' \in \mathbb{R}^{2(K+Z) \cdot d}$ then concatenates \mathbf{T}' and a one-hot vector \mathbf{o}_r (representing the ID of relation r). Here we assume there are V virtual relations representing undiscovered but existent relations in the real world. These virtual relations will have generated meta-tasks. A relation r then can be represented as a one-hot vector $\mathbf{o}_r \in \mathbb{R}^{(|\mathcal{R}_{\text{train}}|+V)}$ to denote its ID.

Decoder. The decoder is to reconstruct the input representation \mathbf{T} with the learned μ and σ , which defines the latent distribution of the given relation r . Given μ and σ , we can sample a latent variable \mathbf{z} from the constructed space $\mathcal{N}(\mu, \sigma)$. Yet the sampling operator makes the model non-differentiable and unable to calculate the gradient. Therefore, we adopt the reparameterization trick (Kingma and Welling, 2014) to solve this issue, which works by $\mathbf{z} = \mu + \sigma \odot \epsilon$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and \odot denotes the element-wise multiplication. The reconstruction should involve the relation information

because the reconstructed task representation is a relation-specific representation. With the sampled \mathbf{z} , the decoder can be denoted as $p_\theta(\mathbf{T}|\mathbf{z}, r)$ and the reconstruction process is defined as follow:

$$\tilde{\mathbf{T}}' = \text{MLP}_{\text{dec}}(\mathbf{z} \oplus \mathbf{o}_r), \quad (10)$$

where $\text{MLP}_{\text{dec}}(\cdot)$ is an L -layer fully connected neural network and $\tilde{\mathbf{T}}' \in \mathbb{R}^{2(K+Z) \cdot d}$ denotes the reconstructed representation of task \mathcal{T} .

With the encoder and decoder, we can define the reconstruction loss to minimize the variational lower bound by:

$$\mathcal{L}_{\text{CVAE}} = -\text{KL}(q_\phi(\mathbf{z}|\mathbf{T}, r)||p_\theta(\mathbf{z}|r)) + \mathbb{E}[\log p_\theta(\mathbf{T}|\mathbf{z}, r)], \quad (11)$$

where the first term is a KL-divergence loss which can be rewritten as $-\text{KL}(q_\phi(\mathbf{z}|\mathbf{T}, r)||p_\theta(\mathbf{z}|r)) = -\frac{1}{2} \sum_{i=1}^M (-\sigma_i + \exp(\sigma_i) + \mu_i^2 - 1)$ by letting the prior distribution $p_\theta(\mathbf{z}|r)$ be $\mathcal{N}(\mathbf{0}, \mathbf{I})$, and the second term is the reconstruction loss defined as $\mathbb{E}[\log p_\theta(\mathbf{T}|\mathbf{z}, r)] = \sum_{i=1}^M \|\tilde{\mathbf{T}}' - \mathbf{T}'\|_2^2$.

4.2.3 Task Generation

With the designed CVAE, we can generate meta-tasks that are related to a relation. We generate tasks for virtual relations that refer to undiscovered relations, which would improve the diversity of tasks and enhance the generalization of the few-shot learner. Note that we do not generate tasks for known relations, because the key issue in low-resource KGC is the absence of representative tasks for a large portion of unknown relations, and those known relations without sufficient triples for meta-task construction. Specifically, given a virtual relation r_v , its one-hot vector representation can be obtained, which is denoted as \mathbf{o}_v . Then we can sample a latent variable \mathbf{z}_v from the prior distribution $p_\theta(\mathbf{z}|r) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. With the sampled \mathbf{z}_v and one-hot vector \mathbf{o}_v , according to Eq.(10), the well-trained decoder can be employed to generate the representation $\tilde{\mathbf{T}}'_v$ for the task corresponding to relation r_v . Then the vector $\tilde{\mathbf{T}}'_v$ can be unsqueezed as $\tilde{\mathbf{T}}_v \in \mathbb{R}^{(K+Z) \times 2d}$, which is further decomposed as four matrices $\widetilde{\mathbf{SH}}_v \in \mathbb{R}^{K \times d}$, $\widetilde{\mathbf{ST}}_v \in \mathbb{R}^{K \times d}$, $\widetilde{\mathbf{QH}}_v \in \mathbb{R}^{Z \times d}$, and $\widetilde{\mathbf{QT}}_v \in \mathbb{R}^{Z \times d}$ standing for the generated representations of head and tail entities in a support set and a query set. And the four matrices form a task $\tilde{\mathcal{T}}_v = \{\widetilde{\mathbf{SH}}_v, \widetilde{\mathbf{ST}}_v, \widetilde{\mathbf{QH}}_v, \widetilde{\mathbf{QT}}_v\}$. The generation process can be repeated multiple times to obtain a set of tasks for a relation, which will be further exploited in the meta-training.

4.3 Task Selector

The generated meta-tasks are leveraged in the meta-training stage with the desire to ease overfitting and improve generalization. However, these tasks inevitably comprise noisy meta-tasks, which cannot represent the corresponding relation well, and would mislead learning of the few-shot learner. To alleviate the adverse impact of noisy tasks, we design a task selector to select beneficial meta-tasks to promote the training of the few-shot learner. Specifically, given a set of generated meta-tasks $\mathcal{T}_{GEN} = \{\tilde{\mathcal{T}}_1, \dots, \tilde{\mathcal{T}}_N\}$, where N denotes the number of tasks, the task selector learns to generate a score to measure the authenticity of a task and decide the beneficial tasks used for meta-training. For a task $\tilde{\mathcal{T}}_i$, the score is calculated as follows:

$$s(\tilde{\mathcal{T}}_i) = \frac{1}{K} \sum_{j=1}^K \text{MLP}_{rs}(\mathbf{e}_{h_j} + \mathbf{r}_{\tilde{\mathcal{T}}_i} - \mathbf{e}_{t_j}), \quad (12)$$

where $\text{MLP}_{rs}(\cdot|\psi)$ is an L -layer fully connected neural network with an output dimension as 1 and parameterized by ψ . Here $\mathbf{r}_{\tilde{\mathcal{T}}_i}$ is obtained using $\widetilde{\mathbf{SH}}_i$ and $\widetilde{\mathbf{ST}}_i$ following Eq.(1), \mathbf{e}_{h_j} denotes the embedding of i -th head entity in $\widetilde{\mathbf{SH}}_i$ and \mathbf{e}_{t_j} is the embedding of i -th tail entity in $\widetilde{\mathbf{ST}}_i$. For better exploration, the task selector adopts a stochastic policy π to make the choice on meta-tasks under the categorical distribution $p = \text{Cat}(\cdot|\mathcal{T}_{GEN})$. The probability $p(\tilde{\mathcal{T}}_i)$ of selecting task $\tilde{\mathcal{T}}_i$ is calculated by $p(\tilde{\mathcal{T}}_i) = s(\tilde{\mathcal{T}}_i) / \sum_{i=1}^N s(\tilde{\mathcal{T}}_i)$.

With the obtained sampling probabilities, we can select B meta-tasks from \mathcal{T}_{GEN} . Then meta-training of few-shot learner $\text{FS}(\cdot)$ can be conducted following Eq.(1) to Eq.(4) using the selected B meta-tasks. Hence the few-shot learner is optimized on B generated meta-tasks as follows:

$$\mathcal{L}_{\widetilde{\text{FS}}} = \sum_{i=1}^B \mathcal{L}_{\tilde{\mathcal{Q}}_i}, \quad (13)$$

where $\mathcal{L}_{\tilde{\mathcal{Q}}_i}$ denotes the loss on generated query set.

To motivate the task selector towards the selection of precise meta-tasks, we evaluate the task selector with feedback signals from the few-shot learner, which reflect the effectiveness of meta-tasks for training the few-shot learner. Specifically, we denote Φ_{OLD} as the parameters of the current few-shot learner and Φ_{NEW} as the parameters after the few-shot learner undertakes a temporary update with Eq.(13). Leveraging $\text{FS}(\cdot|\Phi_{\text{OLD}})$ and

Table 1: Statistics of used Datasets.

KGs	FR	ES	JA	EL
#Entities	13,176	12,382	11,805	5,231
#Relations	178	144	128	111
#Triples	49,015	54,066	28,774	13,839
#Meta-training Tasks	25	30	24	21
#Meta-validation Tasks	5	5	4	4
#Meta-testing Tasks	10	10	8	8

$\text{FS}(\cdot|\Phi_{\text{NEW}})$ and following Eq.(1) to Eq.(4), the reward R is defined as follows:

$$R = \tanh\left(\frac{1}{|\mathcal{T}_{val}|} \sum_{i=1}^{|\mathcal{T}_{val}|} \mathcal{L}_{\mathcal{Q}_i}^{\text{OLD}} - \mathcal{L}_{\mathcal{Q}_i}^{\text{NEW}}\right), \quad (14)$$

where $\mathcal{L}_{\mathcal{Q}_i}^{\text{OLD}}$ and $\mathcal{L}_{\mathcal{Q}_i}^{\text{NEW}}$ are loss functions, defined in Eq.(5) with meta-validation tasks and obtained with $\text{FS}(\cdot|\Phi_{\text{OLD}})$ and $\text{FS}(\cdot|\Phi_{\text{NEW}})$ respectively.

Thereby performance improvement with Φ_{NEW} against Φ_{OLD} will reward the task selector and reinforce it to choose corresponding meta-tasks. To optimize the selector, we adopt the policy gradient algorithm REINFORCE (Williams, 1992) to overcome the issue of non-differentiability of the sampling process. The optimization works by:

$$\psi \leftarrow \psi - \alpha \nabla_{\psi} \log \pi_{\psi}(R - b), \quad (15)$$

where we use π with a slight abuse of notation to denote the task selector parameterized by ψ and α is the learning rate. Besides, b denotes a baseline function, e.g., the moving average of the reward, for reducing computational variance.

4.4 Optimization and Inference

We employ the iterative optimization strategy to optimize three components in the proposed method, i.e., few-shot learner, task generator, and task selector. With the pre-trained representation for entities, the task generator is first optimized to generate meta-tasks \mathcal{T}_{GEN} , then the task selector is applied to conduct selection on \mathcal{T}_{GEN} , next the few-shot learner can be optimized using selected meta-tasks and given meta-tasks \mathcal{T}_{train} . With the obtained reward R , we can update the task selector according to Eq.(15). The whole process is repeated with enough iterations until all components converge.

In the inference stage, we use the optimized few-shot learner to make inferences on new relations in meta-testing set \mathcal{T}_{test} . Similar to the meta-training stage, $\text{FS}(\cdot)$ can learn relation representation using the support set in \mathcal{T}_{test} for a new relation. Then the generated representation is leveraged to evaluate

the test triples in the query set. Note that $\text{FS}(\cdot)$ is not updated by using the query set anymore in the meta-testing stage.

5 Experiments

5.1 Experimental Settings

Dataset. We adopt a multilingual KG dataset (Chen et al., 2020) for evaluation. Specifically, we select four different language-specific KGs extracted from French (FR), Spanish (ES), Japanese (JA) and Greek (EL) DBpedia (Lehmann et al., 2015) as low-resource KGs, as they only have a small number of frequent relations that can offer triples for training task construction. Table 1 shows the statistics of used datasets.

Baselines. To validate the effectiveness of our method, we compare our method with two groups of baseline methods. The first group consists of canonical KG completion models including TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2017), Simple (Kazemi and Poole, 2018), and WRAN (Zhang et al., 2020b). The second group is the few-shot KG completion models, including GMatching (Xiong et al., 2018), FSRL (Zhang et al., 2020a), MetaR (Chen et al., 2019), and GANA (Niu et al., 2021).

Implementation. We implement the proposed method using the Python library Pytorch and conduct all the experiments on an NVIDIA GeForce RTX 3090Ti. Following the popular setting (Zhang et al., 2020a; Xiong et al., 2018), we remove all the relations in \mathcal{T}_{train} and \mathcal{T}_{test} from KG \mathcal{G} and obtain a *background knowledge graph* for pre-training the KG embedding leveraging DistMult (Yang et al., 2015) as the KG encoder. After that, the pre-trained embeddings of entities and relations can be used in the proposed method. Following the popular procedure and the setting in the related works (Zhang et al., 2020a; Xiong et al., 2018), we extract the few-shot learning tasks and divide them into the meta-training, meta-validation, and meta-testing tasks. Specifically, we select the relations with less than 500 but more than 50 triples for preparing the few-shot learning tasks following the popular setting (Zhang et al., 2020a; Xiong et al., 2018). Note that even though the meta-testing tasks include relations with more than 50 triples, we only sample $K = 1$ or $K = 3$ triples as the support set during the meta-testing phase and adopt the rest for evaluation to imitate the real few-shot scenario.

For the few-shot learner, we use three-layer MLP to implement $\text{FS}(\cdot)$ with LeakyReLU as the activation function. The hidden units of each layer in $\text{FS}(\cdot)$ are set as 500, 200, and 100, respectively. And we set the margin parameter γ as 1.0 and set the step size β as 5.0. For the task generator, we use two-layer MLP to implement $\text{MLP}_{enc}(\cdot)$ and $\text{MLP}_{dec}(\cdot)$ with ReLU as the activation function. The hidden units of each layer in $\text{MLP}_{enc}(\cdot)$ are set as 512, and 256, respectively. And the hidden units of each layer in $\text{MLP}_{dec}(\cdot)$ are set as 256, and 512, respectively. Besides, we use a linear layer to implement $f_{\mu}(\cdot)$ and $f_{\sigma}(\cdot)$ with the layer size as 256. For the task selector, we use two-layer MLP to implement $\text{MLP}_{rs}(\cdot)$ with LeakyReLU as the activation function. The hidden units of each layer in $\text{MLP}_{rs}(\cdot)$ are set as 100, and 50, respectively. The dimension of entity embedding is set as 100 for all methods. Besides, we also find the optimal parameters or follow the original paper to achieve the best performance for baseline methods. Note that all triples in the background KGs and the meta-training tasks, as well as all triples corresponding to entity pairs in the support set of meta-validation and meta-testing tasks, should be used to train the canonical KG completion models. For optimization, we employ Adam optimizer to optimize all loss functions with a learning rate of 0.001. The model trained on the meta-training tasks can be used for the meta-validation tasks every 1000 epochs, and the model parameters and corresponding performance will be recorded. Then the model with the best performance on MRR can be used as the final model for meta-testing. Besides, we use early stopping with 30 patient epochs during meta-training. Following the previous work, we report the Hits@1, Hits@5, Hits@10, and MRR (mean reciprocal rank) results to evaluate the performance of few-shot KG completion. Each evaluation is repeated 3 times and averaged results are reported.

5.2 Experimental Results

Performance comparison. The results of all evaluated KG completion models on four different low-resource KGs with $K = 1$ and $K = 3$ are shown in Table 2. We can observe that (1) the proposed method FLOW-KGC has superior performance over canonical KG completion models and few-shot KG completion models by Hits@1, Hits@5, Hits@10, and MRR; (2) The long-tail problem of low-resource KGs has a significant im-

Table 2: Few-shot KG completion comparison on low-resource KGs. The few-shot size $K = 1$ and $K = 3$. The best results are in bold, and the strongest baseline is indicated with *.

(a) The few-shot size $K = 1$.

KGs	ES (Spanish)				EL (Greek)				FR (French)				JA (Japanese)			
	MRR	Hits@1	Hits@5	Hits@10	MRR	Hits@1	Hits@5	Hits@10	MRR	Hits@1	Hits@5	Hits@10	MRR	Hits@1	Hits@5	Hits@10
TransE	0.068	0.003	0.127	0.191	0.043	0.001	0.078	0.124	0.078	0.001	0.159	0.213	0.103	0.008	0.211	0.271
DistMult	0.051	0.019	0.055	0.108	0.059	0.029	0.077	0.116	0.035	0.008	0.045	0.081	0.065	0.008	0.103	0.161
ComplEx	0.089	0.040	0.124	0.188	0.162	0.105	0.217	0.263	0.075	0.036	0.106	0.163	0.091	0.037	0.135	0.188
SimpleE	0.048	0.017	0.054	0.111	0.059	0.02	0.093	0.132	0.029	0.006	0.036	0.068	0.076	0.018	0.111	0.21
WRAN	0.105	0.035	0.165	0.224	0.084	0.053	0.117	0.242	0.097	0.018	0.162	0.223	0.096	0.042	0.135	0.206
GMatching	0.186	0.153	0.248	0.289	0.212	0.146	0.297	0.329	0.168	0.125	0.203	0.274	0.162	0.143	0.258	0.278
GANa	0.195	0.167	0.254	0.265	0.202	0.162	0.294	0.324	0.172	0.128	0.216	0.273	0.167	0.081	0.268*	0.288
FSRL	0.223	0.176	0.265	0.298	0.245	0.176	0.305*	0.347*	0.183	0.145	0.239	0.304	0.194	0.165	0.263	0.297*
MetaR	0.242*	0.194*	0.285*	0.317*	0.247*	0.199*	0.284	0.338	0.210*	0.158*	0.256*	0.315*	0.227*	0.186*	0.264	0.285
Flow-KGC	0.271	0.221	0.311	0.354	0.267	0.210	0.326	0.375	0.240	0.185	0.282	0.348	0.234	0.189	0.278	0.315
Improv. (%)	11.98	13.92	9.12	11.67	8.10	5.52	6.88	8.07	14.28	17.08	10.15	10.47	3.08	1.61	5.30	10.52

(b) The few-shot size $K = 3$.

KGs	ES (Spanish)				EL (Greek)				FR (French)				JA (Japanese)			
	MRR	Hits@1	Hits@5	Hits@10	MRR	Hits@1	Hits@5	Hits@10	MRR	Hits@1	Hits@5	Hits@10	MRR	Hits@1	Hits@5	Hits@10
TransE	0.072	0.003	0.142	0.199	0.081	0.001	0.163	0.228	0.098	0.001	0.202	0.258	0.09	0.007	0.174	0.235
DistMult	0.076	0.028	0.102	0.163	0.149	0.087	0.205	0.272	0.064	0.02	0.089	0.156	0.142	0.074	0.197	0.266
ComplEx	0.114	0.097	0.147	0.235	0.164	0.096	0.237	0.305	0.139	0.076	0.206	0.261	0.137	0.069	0.217	0.27
SimpleE	0.008	0.098	0.118	0.178	0.134	0.069	0.207	0.274	0.058	0.017	0.081	0.145	0.103	0.039	0.158	0.254
WRAN	0.125	0.069	0.174	0.241	0.168	0.082	0.213	0.267	0.125	0.089	0.223	0.244	0.146	0.083	0.187	0.256
GMatching	0.201	0.164	0.252	0.309	0.226	0.164	0.272	0.347	0.175	0.138	0.226	0.292	0.186	0.152	0.263	0.307
GANa	0.206	0.173	0.242	0.252	0.265*	0.208*	0.307	0.389*	0.203	0.158*	0.251*	0.312*	0.151	0.093	0.246	0.310
FSRL	0.234	0.185	0.281	0.327	0.257	0.203	0.310*	0.384	0.182	0.143	0.246	0.284	0.205	0.162	0.269	0.335*
MetaR	0.247*	0.199*	0.297*	0.332*	0.252	0.197	0.302	0.365	0.205*	0.154	0.246	0.297	0.224*	0.167*	0.285*	0.320
Flow-KGC	0.275	0.227	0.320	0.348	0.270	0.217	0.323	0.421	0.223	0.164	0.283	0.345	0.248	0.199	0.292	0.341
Improv. (%)	11.33	14.07	7.74	4.82	1.88	4.32	4.19	8.22	8.78	3.79	12.74	10.57	10.71	19.16	2.45	1.79

impact on canonical KG completion models, such as TransE, DisMult, ComplEx, and SimpleE, which have dramatic performance degradation when inferring the missing triples associated with rare relations; (3) Few-shot KG completion models make progress in improving the inference by pre-training on similar tasks to learn better initialization for rare relations. However, their performance is confined by the deficiency of training tasks; (4) The superior results of FFlow-KGC show the effectiveness of leveraging generated meta-tasks to mitigate the influence of the long-tail problem on low-resource KG completion and improve the generalization.

Ablation study. To gain deeper insight into the effectiveness of each component in the proposed model, we conduct ablation studies by comparing the following variants with FFlow-KGC: (1) FFlow-KGC-FSL that only keeps a few-shot learner and does not adopt the task generator and task selector. (2) FFlow-KGC-w/o-S that adopts the few-shot learner to learn representation and task generator to generate tasks, without the adaptive selection enabled by the task selector. The results on French (FR) and Greek (EL) KG are summarized in Ta-

Table 3: Results of ablation study.

FR	MRR	Hits@1	Hits@5	Hits@10
FFlow-KGC-FSL	0.208	0.160	0.256	0.312
FFlow-KGC-w/o-S	0.231	0.179	0.276	0.337
FFlow-KGC	0.240	0.185	0.282	0.348
EL	MRR	Hits@1	Hits@5	Hits@10
FFlow-KGC-FSL	0.245	0.198	0.285	0.334
FFlow-KGC-w/o-S	0.262	0.205	0.314	0.367
FFlow-KGC	0.267	0.210	0.326	0.375

ble 3, we see that the performance of FFlow-KGC-FSL is inferior to that of others because the variant is a basic few-shot learner and easy to overfit the small set of training tasks. FFlow-KGC-w/o-S outperforms FFlow-KGC-FSL, because the task generator can generate useful synthetic meta-tasks which complement the existing tasks and mitigate the overfitting of the few-shot learner to some extent. FFlow-KGC combining the task generator and task selector achieves the best performance among these variants because the task selector distinguishes beneficial tasks from noisy tasks and utilizes the

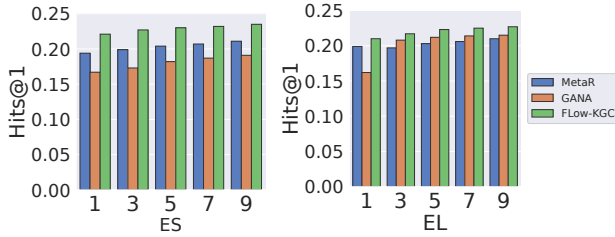


Figure 2: Results on Spanish (ES) and Greek (EL) KGs when varying few-shot size K .

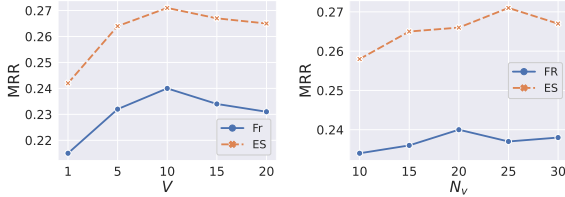


Figure 3: Results on French (FR) and Spanish (ES) KGs when varying the # virtual relations (Left) and # generated tasks for each virtual relation (Right).

selected tasks to learn a better few-shot learner.

Impact of few-shot size K . To investigate the impact of few-shot size K on the performance of few-shot KG completion, we test MetaR, GANA, and FLOW-KGC with size $K = 1, 3, 5, 7, 9$ on Spanish (ES) and Greek (EL) KGs. Figure 2 shows Hits@1 results of three methods with different K . We see that FLOW-KGC consistently outperforms the other two baseline methods with different sizes K , demonstrating the effectiveness of the proposed method. And the performance of all methods gains improvement with K increasing because the larger support set provides more entity pairs to optimize the few-shot learner and learn more accurate representations for rare relations.

Impact of the number of synthetic tasks. The number of virtual relations V and the number of generated tasks N_v for each virtual relation are hyper-parameters to decide the number of synthetic tasks. Figure 3 shows MRR results of FLOW-KGC with different V and N_v on French (FR) and Spanish (ES) KGs. First, with a fixed $N_v = 20$ for French KG and a fixed $N_v = 25$ for Spanish KG, we find that FLOW-KGC has superior performance when $V = 10$. Second, with a fixed $V = 10$, FLOW-KGC with $N_v = 20$ and $N_v = 25$ performs best for French KG and Spanish KG, respectively. We think the reasons behind the observations are similar. A small V or N_v cannot effectively complement the current tasks, and a larger V or N_v ,

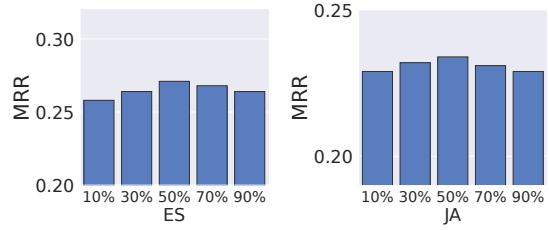


Figure 4: Results on Spanish (ES) and Japanese (JA) KGs when varying the proportion of task selection.

might introduce more noisy tasks and damage the performance of the few-shot learner.

Impact of the proportion of task selection. FLOW-KGC selects B beneficial tasks from the generated tasks to achieve the adaptive selection. Here we denote the selection proportion as $\frac{B}{V \times N_v}$. To evaluate the impact of the proportion of task selection, we evaluate FLOW-KGC with selection proportion 10%, 30%, 50%, 70%, 90%. Figure 4 shows the results on two KGs (ES and JA). We find that FLOW-KGC with a selection proportion of 50% achieves the best performance. We think that a larger proportion unavoidably introduces more noisy tasks into the meta-training phase and a smaller proportion discards extra beneficial tasks, which hurts the effectiveness of FLOW-KGC.

6 Conclusion

In this paper, we proposed a novel few-shot KG completion model to ease the adverse impact of the long-tail issue on low-resource KG completion. Specifically, we designed a task generator based on a conditional variational autoencoder to generate synthetic meta-tasks and proposed a task selector to adaptively select beneficial meta-tasks for optimizing a few-shot learner, which further provides the feedback to update the task selector following the principle of reinforcement learning. These three components constitute our method FLOW-KGC. Extensive experimental results demonstrate the rationality and effectiveness of our proposed method.

Limitations

Despite achieving superior performance, our proposed method requires manual selection for hyper-parameters to decide the number of tasks, i.e., the number of virtual relations V and the number of synthetic tasks N_v for each virtual relation. In future work, we target developing the method with automatic adjustment to add/remove virtual rela-

tions and the corresponding tasks according to the status of the few-shot learner with the training going on by curriculum learning. Besides, although we adopt a task selector to adaptively select beneficial tasks, it is still inevitable to bring noisy tasks in the meta-training stage. We will explore the strategy to achieve better denoising.

References

- Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.
- Stephen Bonner, Ian P Barrett, Cheng Ye, Rowan Swiers, Ola Engkvist, Andreas Bender, Charles Tapley Hoyt, and William Hamilton. 2021. A review of biomedical datasets relating to drug discovery: A knowledge graph perspective. *arXiv preprint arXiv:2102.10062*.
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.
- Yixin Cao, Zhiyuan Liu, Chengjiang Li, Juanzi Li, and Tat-Seng Chua. 2019. Multi-channel graph neural network for entity alignment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1452–1461.
- Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. 2019. Meta relational learning for few-shot link prediction in knowledge graphs. In *EMNLP-IJCNLP*, pages 4217–4226.
- Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *IJCAI*, pages 1511–1517.
- Xuelu Chen, Muhao Chen, Changjun Fan, Ankith Upunda, Yizhou Sun, and Carlo Zaniolo. 2020. Multilingual knowledge graph completion via ensemble knowledge transfer. *Findings of the Association for Computational Linguistics: EMNLP 2020*.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *ACL*, pages 8440–8451.
- Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *AAAI*.
- Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.
- Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M Suchanek. 2015. Fast rule mining in ontological knowledge bases with amie+. *The VLDB Journal*, 24(6):707–730.
- Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*.
- Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 4289–4300.
- Diederik P Kingma and Max Welling. 2014. Stochastic gradient vb and the variational auto-encoder. In *Second International Conference on Learning Representations, ICLR*, volume 19, page 121.
- Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.
- Xin Lv, Yuxian Gu, Xu Han, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2019. Adapting meta knowledge graph information for multi-hop reasoning over few-shot relations. In *EMNLP-IJCNLP*, pages 3376–3381.
- 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. *Communications of the ACM*, 61(5):103–115.
- Tu Dinh Nguyen, Dat Quoc Nguyen, Dinh Phung, et al. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 327–333.
- Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2012. Factorizing yago: scalable machine learning for linked data. In *Proceedings of the 21st international conference on World Wide Web*, pages 271–280.
- Guanglin Niu, Yang Li, Chengguang Tang, Ruiying Geng, Jian Dai, Qiao Liu, Hao Wang, Jian Sun, Fei Huang, and Luo Si. 2021. Relational learning with gated and attentive neighbor aggregator for few-shot knowledge graph completion. *SIGIR*.

- Shichao Pei, Lu Yu, Robert Hoehndorf, and Xiangliang Zhang. 2019a. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *The World Wide Web Conference*.
- Shichao Pei, Lu Yu, Guoxian Yu, and Xiangliang Zhang. 2020. Rea: Robust cross-lingual entity alignment between knowledge graphs. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.
- Shichao Pei, Lu Yu, and Xiangliang Zhang. 2019b. Improving cross-lingual entity alignment via optimal transport. In *International Joint Conference on Artificial Intelligence*.
- Janarthanan Rajendran, Alexander Irpan, and Eric Jang. 2020. Meta-learning requires meta-augmentation. *Advances in Neural Information Processing Systems*, 33:5705–5715.
- Jiawei Sheng, Shu Guo, Zhenyu Chen, Juwei Yue, Li-hong 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 (EMNLP)*, pages 1681–1691.
- Kihyuk Sohn, Honglak Lee, and Xinchun Yan. 2015. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28.
- Zequan Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018a. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*, volume 18, pages 4396–4402.
- Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2018b. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.
- Zhenwei Tang, Shichao Pei, Zhao Zhang, Yongchun Zhu, Fuzhen Zhuang, Robert Hoehndorf, and Xiangliang Zhang. 2022. Positive-unlabeled learning with adversarial data augmentation for knowledge graph completion. *IJCAI*.
- Théo Trouillon, Christopher R Dance, Éric Gaussier, Johannes Welbl, Sebastian Riedel, and Guillaume Bouchard. 2017. Knowledge graph completion via complex tensor factorization. *Journal of Machine Learning Research*, 18:1–38.
- Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2020. [Composition-based multi-relational graph convolutional networks](#). In *International Conference on Learning Representations*.
- Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*.
- 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*, pages 1980–1990.
- Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. *International Conference on Learning Representations*.
- Shiquan Yang, Rui Zhang, and Sarah Erfani. 2020. Graphdialog: Integrating graph knowledge into end-to-end task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1878–1888.
- Chuxu Zhang, Huaxiu Yao, Chao Huang, Meng Jiang, Zhenhui Li, and Nitesh V Chawla. 2020a. Few-shot knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3041–3048.
- Ningyu Zhang, Shumin Deng, Zhanlin Sun, Jiaoyan Chen, Wei Zhang, and Huajun Chen. 2020b. Relation adversarial network for low resource knowledge graph completion. In *Proceedings of The Web Conference 2020*, pages 1–12.
- Wen Zhang, Bibek Paudel, Wei Zhang, Abraham Bernstein, and Huajun Chen. 2019. Interaction embeddings for prediction and explanation in knowledge graphs. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 96–104.
- Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *Thirty-second AAAI conference on artificial intelligence*.
- Zhao Zhang, Fuzhen Zhuang, Hengshu Zhu, Zhiping Shi, Hui Xiong, and Qing He. 2020c. Relational graph neural network with hierarchical attention for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9612–9619.
- Wenxuan Zhou, Fangyu Liu, Ivan Vulić, Nigel Collier, and Muhao Chen. 2021. Prix-lm: Pretraining for multilingual knowledge base construction. *arXiv preprint arXiv:2110.08443*.

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
Section of limitations
- A2. Did you discuss any potential risks of your work?
No potential risks.
- A3. Do the abstract and introduction summarize the paper’s main claims?
Left blank.
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

Left blank.

- B1. Did you cite the creators of artifacts you used?
Not applicable. Left blank.
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C Did you run computational experiments?

Section 5

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 5

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

Section 5

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

Section 5

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

Section 5

D Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

No response.

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

No response.

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

No response.

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

No response.

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

No response.