Few-shot Link Prediction on Hyper-relational Facts

Jiyao Wei[1,2], Saiping Guan[2]*, Xiaolong Jin[1,2]*, Jiafeng Guo[1,2], Xueqi Cheng[1,2]
[1]School of Computer Science and Technology, University of Chinese Academy of Sciences;
[2]Key Laboratory of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Sciences.
{weijiyao20z, guansaiping, jinxiaolong, guojiafeng, cxq}@ict.ac.cn

Abstract

Hyper-relational facts, which consist of a primary triple (head entity, relation, tail entity) and auxiliary attribute-value pairs, are widely present in real-world Knowledge Graphs (KGs). Link Prediction on Hyper-relational Facts (LPHFs) is to predict a missing element in a hyper-relational fact, which helps populate and enrich KGs. However, existing LPHFs studies usually require an amount of high-quality data. They overlook few-shot relations, which have limited instances, yet are common in real-world scenarios. Thus, we introduce a new task, Few-Shot Link Prediction on Hyper-relational Facts (FSLPHFs). It aims to predict a missing entity in a hyper-relational fact with limited support instances. To tackle FSLPHFs, we propose MetaRH, a model that learns Meta Relational information in Hyper-relational facts. MetaRH comprises three modules: relation learning, support-specific adjustment, and query inference. By capturing meta relational information from limited support instances, MetaRH can accurately predict the missing entity in a query. As there is no existing dataset available for this new task, we construct three datasets to validate the effectiveness of MetaRH. Experimental results on these datasets demonstrate that MetaRH significantly outperforms existing representative models.

Keywords: Knowledge graph, hyper-relational facts, few-shot link prediction, knowledge representation

1. Introduction

Link prediction aims to predict a missing element in an incomplete link within KGs. It plays a crucial role in enriching KGs and improving the performance of downstream applications like Web search and question answering (Dong et al., 2015; Lukovnikov et al., 2017). Previous research primarily focuses on binary facts, which are represented as triples (head entity, relation, tail entity). However, real-world KGs often contain hyper-relational facts that involve two entities and several auxiliary attribute-value pairs (Codd, 1983). For instance, more than a third of entities in the popular KG Freebase (Bollacker et al., 2008) are involved in hyper-relational facts (Wen et al., 2016). Therefore, it is essential to extend link prediction beyond binary facts.

In previous approaches to Link Prediction on Hyper-relational Facts (LPHFs), a hyper-relational fact is decomposed into multiple binary facts using virtual entities (Nguyen et al., 2014; Krieger and Willms, 2015). However, this decomposition results in the loss of structure information and increases the number of required parameters, potentially leading to incorrect inferences. To overcome these limitations, recent research has directly modeled hyper-relational facts. Some translation-based approaches (Wen et al., 2016; Zhang et al., 2018) define a hyper-relational fact through an attribute-value mapping. Meanwhile, tensor-based approaches (Fatemi et al., 2019; Liu et al., 2020a) represent the truth space of hyper-relational facts using high-order tensors. More recently, neural network-based approaches (Luo et al., 2023; Wang et al., 2023) have achieved significant performance improvements by leveraging neural networks to capture interactions between elements within hyper-relational facts.

However, current LPHFs methods often overlook the challenge of few-shot relations, even though these relations are prevalent in real-world KGs. For instance, in the benchmark dataset WD50K (Rosso et al., 2020), it is observed that 32.5% of relations have less than 5 instances (see Figure 1). Moreover, real-world KGs are often dynamic, constantly introducing new relations with limited instances. While some existing studies focus on link prediction in few-shot scenarios (Chen et al., 2019; Niu et al., 2021), they are designed for binary facts and cannot deal with attribute-value pairs, which are crucial for fully learning relation representations in hyper-relational facts. Therefore, there is an urgent need for methods that can effectively handle such scenarios.

Thus, we introduce a new task, called Few-Shot Link Prediction on Hyper-relational Facts (FSLPHFs). This task is to predict a missing entity in a hyper-relational fact associated with a relation r, given only a small number of support instances of r (called support set). The main challenge of FSLPHFs lies in effectively learning the representation of r in hyper-relational facts from these limited support instances. We tackle this challenge

* Corresponding authors

LREC-COLING 2024, pages 7196–7207

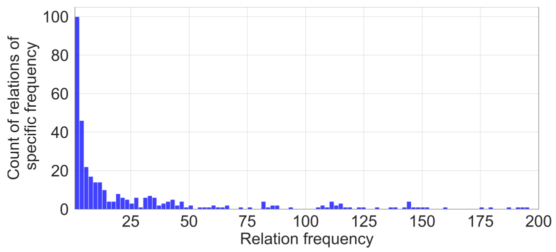

Figure 1: The histogram of relation frequencies in WD50K.

from two perspectives. Firstly, even though the few-shot relations have limited instances, the entities involved have background facts, which can be leveraged to generate few-shot relation representations. Secondly, taking inspiration from the success of meta-learning methods in the field of few-shot learning (Munkhdalai and Yu, 2017; Finn et al., 2017), we can adjust relation representations using loss gradients of the support instances to obtain the essence knowledge of relations, which we refer to as meta relational information.

Based on these considerations, we design a model called MetaRH, which captures meta relational information from limited support instances to predict a missing entity in a query. MetaRH consists of three modules: relation learning, support-specific adjustment, and query inference. The relation learning module generates initial few-shot relation representations by aggregating entity background facts and encoding support instances. The support-specific adjustment module further adjusts relation representations based on the support set to obtain meta relational information. Finally, the query inference module predicts the missing entity in a query using the obtained meta relational information. Due to the lack of datasets designed for FSLPHFs, we construct three datasets based on existing LPHFs benchmark datasets. Through sufficient experiments, we demonstrate that MetaRH significantly outperforms existing models.

In summary, this paper makes the following contributions:

- It propose a new task called Few-Shot Link Prediction on Hyper-relational Facts (FSLPHFs), which is practical in real-world scenarios.
- To tackle FSLPHFs, we propose the MetaRH method, which captures meta relational information from limited support instances to predict the missing entity in a query.
- Three datasets based on existing LPHFs benchmark datasets are constructed, providing valuable resources for evaluating FSLPHFs and further research in this area.
- Through extensive experiments conducted in various settings, we demonstrate that MetaRH achieves superior results in FSLPHFs, showcasing its effectiveness and potential for practical applications.

2. Related Work

Our work is the first to tackle few-shot link prediction on hyper-relational facts, filling a gap in the existing literature. The closest related research areas are Link Prediction on Hyper-relational Facts (LPHFs) and Few-Shot Link Prediction on Binary Facts (FSLPBFs).

2.1. Related Work on LPHFs

Existing LPHFs works can be categorized into three groups: translation-based, tensor-based, and neural network-based.

Translation-based models embed entities and relations into a low-dimensional space and make predictions by translating entities through relations. In the initial work, m-TransH (Wen et al., 2016) represents a hyper-relational fact as a mapping from a sequence of attributes to their corresponding values and models it to obtain the truth value of the fact. RAE (Zhang et al., 2018) enhances m-TransH by considering entity correlations.

Tensor-based models utilize a high-order tensor to represent the truth space of facts and predict new links by reconstructing the tensor. Due to their effectiveness on binary facts, researchers have extended them to handle hyper-relational facts. Some examples of such extensions include m-DistMult (Fatemi et al., 2019), HypE (Fatemi et al., 2019), and GETD (Liu et al., 2020a), which are generalizations of DistMult (Fatemi et al., 2019), SimplE (Kazemi and Poole, 2018), and TuckER (Balažević et al., 2019), respectively.

Neural network-based models utilize neural networks to capture element interactions in hyper-relational facts. NaLP (Guan et al., 2019) represents hyper-relational facts as attribute-value pairs and models their correlation using a fully connected neural network. HINGE (Rosso et al., 2020) and NeuInfer (Guan et al., 2020) represent hyper-relational facts as a primary triple with auxiliary attribute-value pairs and evaluate the validity and compatibility of these two components. StarE (Galkin et al., 2020) proposes a graph representation learning mechanism for hyper-relational facts, enhancing the communication from the auxiliary attribute-value pairs to the primary triple. GRAN (Wang et al., 2021) extends StarE to represent hyper-relational facts as heterogeneous graphs and uses edge-biased attention layers to encode these graphs. Since StarE and GRAN only consider global or local structures in KGs, HAHE (Luo et al., 2023) further proposes a hierarchical attention mechanism that includes global and local attention. ShrinkE (Xiong et al., 2023) extend Box (Abboud et al., 2020) to capture essential inference patterns of hyper-relational facts. The above models generally encode facts in Euclidean space, making it challenging to preserve the hierarchical relationships of entities. PolygonE (Yan et al., 2022) embeds hyper-relational facts as gyro-polygons in hyperbolic poincaré ball and designs a vertex-gyrocentoid optimization goal to measure fact validity. Additionally, HyConvE (Wang et al., 2023) exploits the powerful learning ability of convolutional neural networks for LPHFs.

2.2. Related Work on FSLPBFs

Existing FSLPBFs works can be categorized into two groups: metric learning-based and meta learning-based.

Metric learning-based models match queries to support instances and make predictions based on the match values. GMatching (Xiong et al., 2018) enhances entity representations and learns a matching processor for prediction. FSRL (Zhang et al., 2020) extends GMatching by integrating information from multiple instances rather than relying on just one. FAAN (Sheng et al., 2020) further introduces an adaptive attention mechanism that selectively focuses on entity properties.

Meta learning-based models calculate the gradient on support instances and quickly optimize parameters. MetaR (Chen et al., 2019) transfers relation information from support instances to queries using relation gradients. MetaP (Jiang et al., 2021) utilizes more efficient convolutional filters and proposes a validity balance mechanism of negative samples. GANA (Niu et al., 2021) combines MAML (Finn et al., 2017) and TransH (Wang et al., 2014) to predict few-shot complex relations.

Existing LPHFs models overlook few-shot relations, while FSLPBFs models focus on binary facts and cannot handle hyper-relational facts.

3. Problem Formulation

In this section, we provide the definitions of hyper-relational facts, link prediction on hyper-relational facts, and few-shot link prediction on hyper-relational facts in turn.

Definition 1 ***Hyper-relational facts*** *are composed of a primary triple* (h, r, t) *and several auxiliary attribute-value pairs* $\{(a_i, v_i)\}_{i=1}^{m}$ *(Rosso et al., 2020). Here,* $r, a_i, ..., a_m \in R$ *and* $h, t, v_1, ..., v_m \in E$*, with* m *denoting the number of auxiliary attribute-value pairs,* E *representing the set of entities and values, and* R *denoting the set of relations and attributes.*

For instance, the hyper-relational fact, *Einstein* studied for a *Doctorate's Degree* of *Physics* at *University of Zurich* from *1901* to *1905*, can be represented as:

((Einstein, studied for, Doctorate's Degree),{
(major, Physics),
(university, the University of Zurich),
(begin-time, 1901), (end-time, 1905)}).

Definition 2 ***Link Prediction on Hyper-relational Facts (LPHFs)*** *aims to predict one missing element in a hyper-relational fact (Wen et al., 2016), such as predicting the tail entity of the incomplete hyper-relational fact* $((h, r, ?), \{(a_i, v_i)\}_{i=1}^{m})$.

Definition 3 ***Few-Shot Link Prediction on Hyper-relational Facts (FSLPHFs)*** *aims to predict a missing entity*[1] *in a query in the query set* $\mathcal{Q}_r = \{((h_q, r, ?), \{(a_{q_i}, v_{q_i})\}_{i=1}^{m})\}$ *of a few-shot relation* r*, with* k *support instances* $\mathcal{S}_r = \{((h_s^j, r, t_s^j), \{(a_{s_i}^j, v_{s_i}^j)\}_{i=1}^{m})\}_{j=1}^{k}$ *(referred to as the support set) given, called* k*-shot link prediction on hyper-relational facts.*

The training process of FSLPHFs is based on a set of tasks, wherein each task is associated with a few-shot relation and has its support set and query set. Besides, each task has an entity candidate set, which contains candidate entities that satisfy possible entity types, following (Xiong et al., 2018). The testing process is performed on a set of new tasks, wherein each task is associated with a few-shot relation that has not appeared in the training process and has its support set and query sets.

Finally, we assume that the method has access to background data $\mathcal{B}$, which contains background facts about entities in $\mathcal{S}_r$, following (Xiong et al., 2018). The background facts of an entity e is a set of facts with e as the head entity. To utilize $\mathcal{B}$ fully, inverse facts $\{((t, r^{-1}, h), \{(a_i, v_i)\}_{i=1}^{m})\}$ are added to $\mathcal{B}$. For example, for a fact in the background data, ((Game of Thrones (Q23572), cast member (P161), Ciarán Hinds (Q314892)), {(character role (P453), Mance Rayder (Q5991029))}), we add its inverse facts ((Ciarán Hinds (Q314892), cast member (P161)$^{-1}$, Game of Thrones (Q23572)), {(character role (P453), Mance Rayder (Q5991029))}) to the background data. To avoid data leakage, no few-shot relations exist in $\mathcal{B}$.

[1] In this paper, we conduct FSLPHFs through tail entity prediction following (Xiong et al., 2018; Galkin et al., 2020), as head entity prediction can be transformed into tail entity prediction easily through inverse relations.

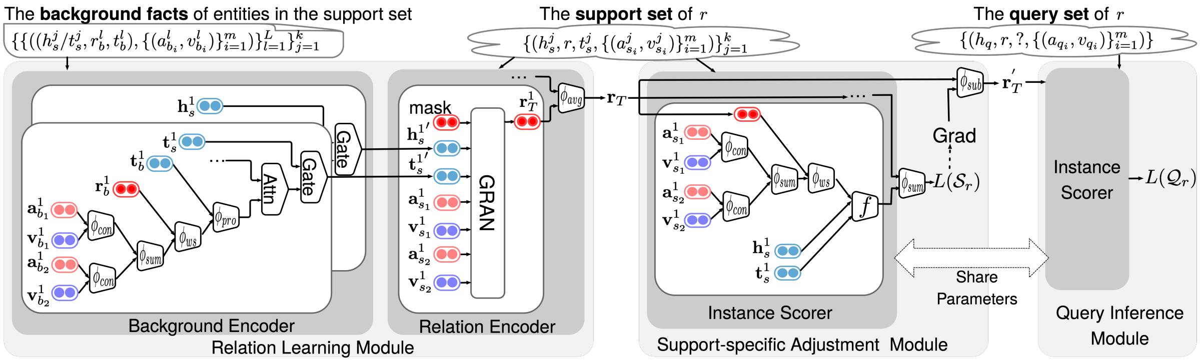

Figure 2: The overview of MetaRH model. To distinguish the different elements in hyper-relational facts, we use red for relations, blue for head and tail entities, pink for attributes, and purple for values.

4. The Proposed MetaRH Model

We propose MetaRH to tackle FSLPHFs. MetaRH consists of three modules: relation learning, support-specific adjustment, and query inference, as illustrated in Figure 2. To be clear, in Figure 2 and what follows, we illustrate MetaRH with the background facts = $\{\{((h_s^j/t_s^j, r_b^l, t_b^l), \{(a_{b_i}^l, v_{b_i}^l)\}_{i=1}^m)\}_{l=1}^L\}_{j=1}^k$, support set $\mathcal{S}_r$ = $\{((h_s^j, r, t_s^j), \{(a_{s_i}^j, v_{s_i}^j)\}_{i=1}^m)\}_{j=1}^k$, and query set $\mathcal{Q}_r$ = $\{((h_q, r, ?), \{(a_{q_i}, v_{q_i})\}_{i=1}^m)\}$.

4.1. Relation Learning Module

It is designed to obtain an initial representation of few-shot relation r using a background encoder and a relation encoder.

The background encoder utilizes background facts to generate semantic-rich entity representations in the support set. To leverage priori knowledge from the background facts and enhance entity representations, we employ a Graph Neural Network with attention and gating mechanisms.

For example, the semantic-rich representation $\mathbf{t}_s^{j\prime}$ of entity t_s^j is obtained by combining the initial entity representation $\mathbf{t}_s^j$ with its background fact representation $\mathbf{b}^l$, using attention values α and a gate value g, as follows:

$$\mathbf{t}_s^{j\prime} = \sigma\left(\sum_{l=1}^{L} g\alpha^l \mathbf{b}^l + (1-g)\,\mathbf{t}_s^j\right), \tag{1}$$

where L is the number of background facts per entity; the background facts of $\mathbf{t}_s^j$ is a set of facts in $\mathcal{B}$ with $\mathbf{t}_s^j$ as the head entity; α^l represents the attention value of background fact b^l; σ is an activation function. The semantic-rich head entity representation $\mathbf{h}_s^{j\prime}$ is obtained in the same manner as $\mathbf{t}_s^{j\prime}$. Next, we provide a detailed explanation of the background fact representation $\mathbf{b}^l$, attention value α^l, and gate value g.

The background fact representation $\mathbf{b}^l$ is calculated by aggregating all the elements in b^l. This involves using a weighted sum operation ϕ_{ws} and a project operation ϕ_{pro_2} to fuse the auxiliary attribute-value pairs representation $\mathbf{q}_b^l$ to the relation representation $\mathbf{r}_b^l$. The new relation representation is then combined with the tail entity representation $\mathbf{t}_b^l$ using a project operation ϕ_{pro} and a concatenate operation ϕ_{con}. As a result, $\mathbf{b}^l$ is obtained as follows:

$$\begin{aligned}\mathbf{b}^l &= \phi_{pro}(\phi_{con}(\phi_{ws}(\mathbf{r}_b^l, \phi_{pro_2}(\mathbf{q}_b^l)), \mathbf{t}_b^l)) \\ &= \mathbf{W}_1\left[(\tau \odot \mathbf{r}_b^l + (1-\tau) \odot \mathbf{W}_2\mathbf{q}_b^l); \mathbf{t}_b^l\right] + \mathbf{b}_1,\end{aligned} \tag{2}$$

where b^l contains $((t_s^j, r_b^l, t_b^l),\ \{(a_{b_i}^l, v_{b_i}^l)\}_{i=1}^m)$; $\mathbf{W}_1$ and $\mathbf{W}_2$ are parameterized projection matrixes; $\mathbf{b}_1$ is a parameterized bias; τ is a relation weight hyper-parameter; $\odot$ is a scalar product operation; inspired by Galkin et al. (Galkin et al., 2020), $\mathbf{q}_b^l$ is obtained using a position invariant summation function ϕ_{sum} and a rotate function ϕ_{rot} (Sun et al., 2019), as follows:

$$\begin{aligned}\mathbf{q}_b^l &= \phi_{sum}(\{\phi_{rot}(\mathbf{a}_{b_i}^l, \mathbf{v}_{b_i}^l)\}_{i=1}^m) \\ &= \sum_{(\mathbf{a}_{b_i}^l, \mathbf{v}_{b_i}^l) \in \{(\mathbf{a}_{b_i}^l, \mathbf{v}_{b_i}^l)\}_{i=1}^m} \phi_{rot}(\mathbf{a}_{b_i}^l, \mathbf{v}_{b_i}^l),\end{aligned} \tag{3}$$

where $\mathbf{a}_{b_i}^l$ and $\mathbf{v}_{b_i}^l$ are the corresponding representations of $a_{b_i}^l$ and $v_{b_i}^l$, respectively.

To capture the most valuable information from background facts and filter out noisy facts, attention mechanisms and gating mechanisms are employed, referring to (Sheng et al., 2020; Niu et al., 2021). The attention value α^l of background fact b^l is calculated by applying the softmax function on all absolute attention values, as follows:

$$\alpha^l = \frac{\exp(d^l)}{\sum_{l=1}^{L} \exp(d^l)}. \tag{4}$$

$$d^l = \mathrm{LeakyReLU}\left(\mathbf{U}_1^T \mathbf{b}^l\right), \tag{5}$$

where d^l is the absolute attention value of b^l; LeakyReLU$(\cdot)$ (Maas et al., 2013) is an activation function; $\mathbf{U}_1$ is a weight vector.

To further filter out noisy background facts, gating mechanisms are implemented. The gate value g of all background facts is calculated using a sigmoid function as follows:

$$g = \text{sigmoid}\left(\mathbf{U}_2^T \sum_{l=1}^{L} \alpha^l \mathbf{b}^l + b_g\right), \tag{6}$$

where $\mathbf{U}_2$ is a weight vector, and b_g is a scalar bias.

The relation encoder aims to generate few-shot relation representations with semantic-rich entity representations as input.

Many existing LPHFs models can generate relation representations based on hyper-relational facts. In this work, we select GRAN (Wang et al., 2021) as the relation encoder, since it has shown effectiveness in the LPHFs task. Taking the support instance $((h_s^j,\ r,\ t_s^j),\ \{(a_{s_i}^j, v_{s_i}^j)\}_{i=1}^m)$ as an example, the semantic-rich instance representation, $((\mathbf{h}_s^{j'},\ mask,\ \mathbf{t}_s^{j'}),\ \{(\mathbf{a}_{s_i}^j, \mathbf{v}_{s_i}^j)\}_{i=1}^m)$, is represented as a heterogeneous graph $\mathcal{G}$, where $mask$ is a special token denoting the few-shot relation r. More details on GRAN can be found in (Wang et al., 2021). Then graph $\mathcal{G}$ is then processed through a stack of D GRAN blocks, as follows:

$$\mathcal{G}^d = \text{GRAN}\left(\mathcal{G}^{d-1}\right), d = 1, 2, \cdots, D, \tag{7}$$

where $\mathcal{G}^d$ is the hidden state after d-th layer. The representation of $mask$ in the last layer is selected as the few-shot relation representation $\mathbf{r}_{T_j}$.

The relation representations of other support instances are obtained similarly. The few-shot relation representation $\mathbf{r}_T$ of the current task is obtained through an average operation ϕ_{avg} :

$$\mathbf{r}_T = \phi_{avg}(\{\mathbf{r}_{T_j}\}_{j=1}^k) = \frac{1}{k}\sum_{j=1}^{k} \mathbf{r}_{T_j}, \tag{8}$$

where k is the number of support instances.

4.2. Support-specific Adjustment Module

The previous module generates a few-shot relation representation. However, it is coarse due to the simple aggregation operation in Equation 8. Thus, the support-specific adjustment module is designed to obtain meta relational information that represents common knowledge within the task. This module utilizes the gradient on support instances to guide the adjustment of the coarse relation representation based on an instance scorer. Before introducing the adjustment of the relation representation, we first introduce the instance scorer and the loss of the support set.

Algorithm 1 The training process of MetaRH.

Input: Training tasks $\mathcal{T}_{training}$; Initial parameters.

1: **repeat**
2: Sample mini-batch tasks $\mathcal{F}_t$ from $\mathcal{T}_{training}$.
3: **for** each task in $\mathcal{F}_t$ **do**
4: Sample few-shot instances as S_r.
5: Sample a batch of instances as Q_r.
6: Get the background facts of S_r.
7: Generate semantic-rich entity representations in S_r (Equation 1∼Equation 6).
8: Generate an initial few-shot relation representation $\mathbf{r}_T$ (Equation 7∼Equation 8).
9: Calculate the loss of $\mathcal{S}_r$ (Equation 9∼Equation 12).
10: Generate meta relational information $\mathbf{r}'_T$ (Equation 13).
11: Calculate the loss of $\mathcal{Q}_r$ (Equation 14).
12: **end for**
13: Update model parameters.
14: **until** process completes maximum times.

The instance scorer evaluates the semantic connections between few-shot relations and other elements in instances. Previous research in FSLPBFs (Chen et al., 2019) has shown that the translation-based model TransE (Bordes et al., 2013) performs well as an instance scorer. Therefore, we adopt a translation-based instance scorer in this work, which is designed as follows:

Taking the support instance $((h_s^j, r, t_s^j),\ \{(a_{s_i}^j, v_{s_i}^j)\}_{i=1}^m)$ as an example, we first calculate the auxiliary attribute-value pairs representation $\mathbf{q}_s$ similarly as $\mathbf{q}_b^l$ (see Equation 3) and aggregate it to $\mathbf{r}_T$, as follows:

$$\begin{aligned}\mathbf{q}_s^l &= \phi_{sum}(\{\phi_{rot}(\mathbf{a}_{s_i}^l, \mathbf{v}_{s_i}^l)\}_{i=1}^m) \\ &= \sum_{(\mathbf{a}_{s_i}^l, \mathbf{v}_{s_i}^l)\in\{(\mathbf{a}_{s_i}^l, \mathbf{v}_{s_i}^l)\}_{i=1}^m} \phi_{rot}(\mathbf{a}_{s_i}^l, \mathbf{v}_{s_i}^l),\end{aligned} \tag{9}$$

$$\mathbf{r}_s^j = \phi_{ws}(\mathbf{r}_T, \phi_{pro_2}(\mathbf{q}_s^j)) = \tau \odot \mathbf{r}_T + (1-\tau) \odot \mathbf{W}_2\mathbf{q}_s^j, \tag{10}$$

where $\mathbf{a}_{s_i}^j$, $\mathbf{v}_{s_i}^j$ are the corresponding representations of $a_{s_i}^j$, $v_{s_i}^j$ respectively; $\mathbf{r}_s^j$ is the new relation representation that incorporates the representation of auxiliary attribute-value pairs.

Then, following TransE, the score of the support instance is calculated as follows:

$$f\left(\mathbf{h}_s^j, \mathbf{r}_s^j, \mathbf{t}_s^j\right) = \left\|\mathbf{h}_s^j + \mathbf{r}_s^j - \mathbf{t}_s^j\right\|, \tag{11}$$

where $\mathbf{h}_s^j$, $\mathbf{t}_s^j$ are the corresponding representations of h_s^j, t_s^j respectively; $||\mathbf{x}||$ is the L2 norm of vector $\mathbf{x}$; this function is denoted as "f" in Figure 2. To obtain accurate common knowledge from the support set, the initial entity representations are used instead of rich-semantic ones.

The loss of support set is defined as follows:

$$L\left(\mathcal{S}_r\right)=\sum_{\left(\mathbf{h}_s^j,\mathbf{r}_s^j,\mathbf{t}_s^j\right)\in\mathcal{S}_r}\left[\mu+f\left(\mathbf{h}_s^j,\mathbf{r}_s^j,\mathbf{t}_s^j\right)-f\left(\mathbf{h}_s^j,\mathbf{r}_s^j,\mathbf{t}_s^{j''}\right)\right]_+, \tag{12}$$

where $[x]_+ = max[0,x]$ is hinge loss; μ is a margin hyper-parameter; $\mathbf{t}_s^{j''}$ is generated by randomly corrupting tail entities of support instances.

The adjustment of $\mathbf{r}_T$ is guided by the gradient of $\mathbf{r}_T$, which indicates how $\mathbf{r}_T$ should be adjusted, as $L(\mathcal{S}_r)$ represents the ability of the instance scorer to encode the truth values of instances. Therefore, we obtain meta relational information $\mathbf{r}'_T$, as follows:

$$\begin{aligned}\mathbf{r}'_T &= \phi_{sub}(\mathbf{r}_T, Grad(\mathbf{r}_T)) \\ &= \mathbf{r}_T - \beta\frac{\mathrm{d}L\left(\mathcal{S}_r\right)}{\mathrm{d}\mathbf{r}_T},\end{aligned} \tag{13}$$

where ϕ_{sub} is a subtraction operation; $Grad(\mathbf{r}_T)$ is the gradient of $\mathbf{r}_T$; β indicates the step size of the gradient when adjusting $\mathbf{r}_T$.

4.3. Query Inference Module

The query inference module predicts the missing entity in a query using an instance scorer. To enhance the training efficiency of MetaRH, the instance scorer in the query inference module adapts the same structure and shares parameters as the instance scorer introduced in the support-specific adjustment module (see Section 4.2).

The loss of query set is computed similarly to $L(\mathcal{S}_r)$ (see Equation 12), as follows:

$$L(\mathcal{Q}_r)=\sum_{(\mathbf{h}_q,\mathbf{r}_q,\mathbf{t}_q)\in\mathcal{Q}_r}\left[\mu+f(\mathbf{h}_q,\mathbf{r}_q,\mathbf{t}_q)-f\left(\mathbf{h}_q,\mathbf{r}_q,\mathbf{t}''_q\right)\right]_+, \tag{14}$$

where $\mathbf{h}_q$, $\mathbf{t}_q$ is the corresponding representations of h_q, t_q respectively; the new relation representation $\mathbf{r}_q$ is obtained by combining the representation of attribute-value pairs and meta relational information, similar to $\mathbf{r}_s^j$ (see Equation 10); $\mathbf{t}''_q$ is the negative entity representation, which is generated in a similar way as $\mathbf{t}_s^{j''}$ (see Equation 12). For further details on the training process, refer to Algorithm 1.

5. Experiments

5.1. Datasets

Since there is no dataset specifically designed for FSLPHFs, we construct three new datasets, F-WikiPeople, F-JF17K, and F-WD50K, by modifying existing LPHFs benchmark datasets WikiPeople (Guan et al., 2018), JF17K (Wen et al., 2016), and WD50K (Rosso et al., 2020), respectively. These LPHFs datasets are derived from real-world KGs and are widely used in LPHFs. Specifically, the JF17K dataset is derived from Freebase (Bollacker et al., 2008), and the Wikipeople and WD50K datasets is derived from Wikidata (Vrandečić and Krötzsch, 2014). The Wikipeople dataset stores a large number of facts related to people, while WD50K stores a large number of facts in which head entities appear in the well-known knowledge graph FB15K-237 (Bordes et al., 2013). We believe that our proposed datasets will provide valuable resources for further research in this field.

New FSLPHFs datasets are constructed as follows:

- Select relations with 20-1000 instances[2] as few-shot relations from each existing dataset.
- Get few-shot data by retrieving the instances of few-shot relations.
- Remove instances with few-shot relations in auxiliary attribute-value pairs from the few-shot data to prevent data leakage.
- Get background data $\mathcal{B}$ by retrieving the instances of entities in the few-shot data from the original dataset.
- Remove instances containing few-shot relations from $\mathcal{B}$ to prevent data leakage.
- Divide the few-shot data into training tasks $\mathcal{T}_{training}$, validation tasks $\mathcal{T}_{validation}$, and testing tasks $\mathcal{T}_{testing}$, in the proportion of 85%: 5%: 10%, following (Xiong et al., 2018).

Table 1 provides statistics of the constructed datasets, including counts for various elements: #X is the number of X, E-q and R-q denote the values and attributes in auxiliary attribute-value pairs respectively, B-facts and F-facts denote the facts in background data and few-shot data respectively, B-N-rate and F-N-rate denote the proportion of hyper-relational facts in background data and few-shot data respectively, and Tasks denotes the few-shot tasks.

5.2. Experimental Settings

Baselines. Due to the lack of models designed specifically for FSLPHFs, MetaRH is primarily compared with LPHFs and FSLPBFs models: (1) representative or state-of-the-art LPHFs models: m-TransH (Wen et al., 2016), HypE (Fatemi et al., 2019), NeuInfer (Guan et al., 2020), HINGE (Rosso et al., 2020), StarE (Galkin et al., 2020), GRAN (Wang et al., 2021), PolygonE (Yan

[2]The lower boundary is to have enough facts for evaluation. The upper boundary is to retain some facts to be used as background data.

Dataset	#E	#R	#E-q	#R-q	#B-facts	B-N-rate	#F-facts	F-N-rate	#Tasks
F-WikiPeople	40529	237	4663	75	314670	9.1%	4470	1.5%	30
F-JF17K	19721	480	4928	127	86415	49.3%	5157	19.2%	52
F-WD50K	43802	697	10242	85	358439	13.8%	21214	1.8%	118

Table 1: Statistics of the constructed datasets.

Method	F-WikiPeople				F-JF17K				F-WD50K			
	MRR	Hits@10	Hits@5	Hits@1	MRR	Hits@10	Hits@5	Hits@1	MRR	Hits@10	Hits@5	Hits@1
m-TransH	0.197	0.403	0.309	0.101	0.045	0.076	0.047	0.021	0.051	0.081	0.047	0.015
HypE	0.291	0.516	0.368	0.195	0.042	0.111	0.040	0.014	0.051	0.106	0.063	0.024
PolygonE	0.231	0.388	0.268	0.137	0.057	0.192	0.134	0.025	0.052	0.107	0.060	0.018
NeuInfer	0.289	0.581	0.455	0.155	0.092	0.156	0.111	0.061	0.133	0.231	0.180	0.078
HINGE	0.333	0.439	0.276	0.277	0.084	0.124	0.095	0.064	0.154	0.267	0.213	0.089
StarE	0.286	0.558	0.471	0.118	0.117	0.151	0.135	0.090	0.102	0.177	0.134	0.057
GRAN	0.287	0.432	0.374	0.209	0.119	0.157	0.124	0.101	0.126	0.222	0.162	0.077
ShrinkE	0.314	0.504	0.421	0.221	0.051	0.123	0.063	0.020	0.046	0.081	0.059	0.024
HyConvE	0.364	$\underline{0.621}$	0.428	0.272	0.177	0.289	0.234	$\underline{0.123}$	0.086	0.176	0.118	0.038
HAHE	$\underline{0.392}$	0.583	0.480	$\underline{0.306}$	$\underline{0.182}$	$\underline{0.293}$	$\underline{0.276}$	0.117	0.157	0.265	0.206	0.102
FAAN	0.266	0.550	0.412	0.092	0.032	0.090	0.034	0.003	0.116	0.226	0.166	0.059
MetaR	0.282	0.556	0.459	0.147	0.047	0.086	0.055	0.022	0.108	0.183	0.139	0.064
GANA	0.341	0.475	0.371	0.275	0.074	0.218	0.130	0.016	$\underline{0.176}$	0.313	0.246	0.100
ChatGPT	-	0.584	**0.548**	**0.358**	-	0.165	0.140	0.093	-	**0.548**	**0.474**	**0.237**
MetaRH	**0.415**	**0.644**	$\underline{0.500}$	$\underline{0.318}$	**0.214**	**0.329**	**0.292**	**0.141**	**0.192**	$\underline{0.340}$	$\underline{0.278}$	$\underline{0.109}$

Table 2: Few-shot link prediction performance on hyper-relational facts.

et al., 2022), ShrinkE (Xiong et al., 2023), HyConvE (Wang et al., 2023), and HAHE (Luo et al., 2023); (2) advanced FSLPBFs models: MetaR (Chen et al., 2019), FAAN (Sheng et al., 2020), and GANA (Niu et al., 2021). More details on these models can be found in Section 2. Additionally, MetaRH is compared with ChatGPT [3], a recently prominent Large Language Model (LLM).

Evaluation metrics used are Hits@k and Mean Reciprocal Rank (MRR), with k = 1, 5, 10, following (Xiong et al., 2018). The hits@k metric is the proportion of the correct answer ranked within the top k, while the MRR metric is the average of the reciprocal rank of the correct answer. Higher values of MRR and Hits@k indicate better performance.

Implementation details. Hyper-parameters of MetaRH are selected within the following ranges: The embedding dimension $\in \{50, 100\}$, the batch size of tasks per epoch $\in \{128, 256, 512, 1024, 2048\}$, the batch size of queries per task $\in \{1, 2, 3, 4, 5\}$, the learning rate $\in \{5e{-}3, 1e{-}3, 5e{-}4, 1e{-}4\}$, the maximum number of background facts per entity $\in \{10, 20, 30, 50\}$, the margin $\mu \in \{1, 2, 3, 4, 5\}$, and the relation weight $\tau \in [0.0, 1.0]$ with step is 0.1. The embedding layer is initialized with pre-trained embeddings trained on the background data with HINGE, following (Xiong et al., 2018). The Adam optimizer (Kingma and Ba, 2014) is used to optimize the model. We conduct all experiments in the 5-shot scenario, that is, k is set to 5. The Code and datasets of this paper can be found at https://github.com/JiyaoWei/MetaRH.

To ensure a fair comparison, LPHFs baselines are trained using all facts in $\{\mathcal{T}_{training}, \mathcal{S}_r \in \mathcal{T}_{validation} \cup \mathcal{T}_{testing}, \mathcal{B}\}$, while FSLPBFs baselines are trained using the same $\mathcal{T}_{training}$ and $\mathcal{B}$ employed by MetaRH. The hyper-parameters of all baselines, except ChatGPT, are tuned on each experimental dataset.

For ChatGPT, to enhance the persuasiveness of the experiments, we manually constructed the prompt of ChatGPT for each query following Zhu et al. (Zhu et al., 2023). Furthermore, we modified the prompt to produce multiple candidates for a more in-depth comparative analysis. Specifically, we added "Please list the 10 most likely answers and rank them in descending order of confidence." at the end of the current prompt. ChatGPT would generate 10 rows, each representing one candidate answer, i.e. "1. {candidate}\n 2. {candidate}\n ... 10. {candidate}". {candidate} indicates a generated answer. For the metric Hits@10, if the true entity appears in any of the answers generated by ChatGPT, the prediction is considered correct. For the metric Hits@k, if the true entity appears in the first k generated answers, the prediction is considered correct. Figure 3 illustrates the format of the prompts, which include support instances and a query. Given that the accuracy of the responses generated by ChatGPT is the only metric available, we only show its Hits metrics.

5.3. Experimental Results and Analysis

The experimental results for all three datasets are displayed in Table 2, highlighting the best results in bold and the second-best results underlined. We have the following observations:

Comparing MetaRH with LPHFs baselines, MetaRH outperforms all existing models across all three datasets. For instance, in terms of the Hits@10 metric, MetaRH achieves improvements

[3] https://openai.com/blog/chatgpt/

Methods	F-WikiPeople MRR	Hits@10	Hits@5	Hits@1	F-JF17K MRR	Hits@10	Hits@5	Hits@1	F-WD50K MRR	Hits@10	Hits@5	Hits@1
MetaRH	**0.415**	**0.644**	**0.500**	**0.318**	**0.214**	**0.329**	**0.292**	**0.141**	**0.192**	**0.340**	**0.278**	**0.109**
-background	0.382	0.540	0.421	0.304	0.199	0.306	0.285	0.109	0.177	0.326	0.265	0.091
-adjustment	0.328	0.450	0.414	0.261	0.182	0.299	0.270	0.104	0.152	0.285	0.219	0.082

Table 3: Experimental results of the ablation study.

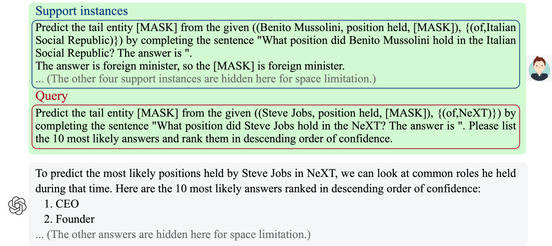

Figure 3: An example of ChatGPT prompts for FSLPHFs.

of 2.3% on the F-WikiPeople dataset (3.7% relative improvement), 3.9% on the F-JF17K dataset (13.3% relative improvement), and 7.3% on the F-WD50K dataset (27.3% relative improvement). The remarkable success of MetaRH can be attributed to learning meta relational information. This targeted design empowers MetaRH to effectively capture the essential knowledge of relations with limited instances. Notably, MetaRH demonstrates the most significant performance improvement on the F-WD50K dataset, which involves the maximum number of training tasks. This observation suggests that the more training tasks there are, the stronger MetaRH's ability to learn meta relational information.

Comparing MetaRH with FSLPBFs baselines, MetaRH outperforms existing models across all three datasets. For instance, in terms of the Hits@10 metric, MetaRH achieves improvements of 8.8% on F-Wikidata (15.8% relative improvement), 11.1% on the F-JF17K dataset (50.9% relative improvement), and 2.7% on the F-WD50K dataset (8.6% relative improvement). This improvement demonstrates the effectiveness of leveraging auxiliary attribute-value pairs in few-shot relation learning. Moreover, MetaRH achieves a significant improvement on the F-JF17K dataset, which has a high proportion of hyper-relational facts, further emphasizing the importance of using auxiliary attribute-value pairs.

Comparing MetaRH with ChatGPT, MetaRH performs better on the F-JF17K dataset but falls short on the F-WikiPeople and F-WD50K datasets, considering most metrics. This performance discrepancy may be due to the variance in data sources. The F-JF17K dataset is derived from Freebase, while the F-WikiPeople and F-WD50K datasets are derived from Wikidata. MetaRH achieved a relative improvement of up to 51.6% in the metric hits@1 on the F-JF17K dataset, where 87% of the knowledge is domain-specific knowledge, including film and sport. MetaRH does not perform well on the F-WikiPeople dataset and F-WD50K dataset, since these two datasets are derived from Wikidata, storing a large amount of generalized domain knowledge such as geography, country, etc. This indicates that knowledge graph models are still necessary in the real scenario currently. They achieve better performance on the reasoning task in non-generalized domains, as demonstrated by the LLM survey (Pan et al., 2024). We also speculate that the unpublished training datasets used by ChatGPT include Wikidata or related datasets such as Wikipedia, but not Freebase. The opacity of the training data seriously affects its practical applications. Additionally, crafting high-quality prompts is crucial, but it is laborious and requires expert experience. For the metric MRR, calculating it of ChatGPT on link prediction is still a challenge since we can only check if the answer is in the response of ChatGPT but is almost impossible to get the rank for each answer.

Furthermore, we conduct experiments on the largest dataset F-WD50K to analyze the impact of the k-shot setting. We follow Sheng et al. (Sheng et al., 2020) to vary k from 1 to 6. We compared MetaRH with several competitive baseline models, namely HINGE, NeuInfer, GANA, and HAHE. The results, depicted in Figure 4, show that MetaRH consistently outperforms the baseline models across various k values. Additionally, the performance of baselines does not plateau. We speculate that it is due to that they are less capable of data utilization. Specifically, the baselines for link prediction on hyper-relational facts (e.g., HINGE, Nueinfer, and HAHE) cannot effectively capture the essential knowledge of relations with limited instances. The few-shot link prediction baseline (e.g., MetaR) cannot leverage auxiliary attribute-value pairs. They all have data utilization issues and are sensitive to different values of k.

5.4. Ablation Studies

The two essential components of MetaRH are the background encoder and support-specific adjustment module. To evaluate their necessity, ablation studies are conducted on all three datasets. The results in Table 3 provide important insights. Firstly, removing the background encoder (-background) results in a noticeable performance

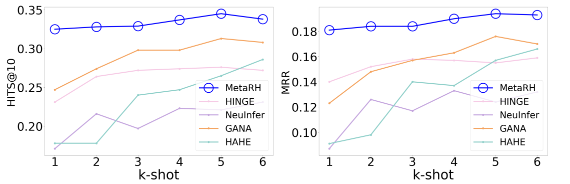

Figure 4: Impact of the few-shot size on the F-WD50K dataset.

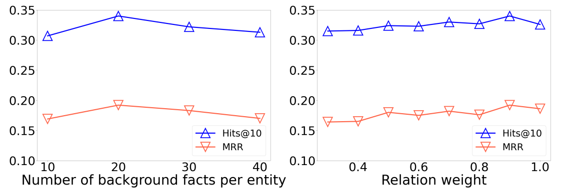

Figure 5: Impact of the number of background facts per entity and relation weight on the F-WD50K dataset.

drop. This highlights the benefit of enhancing entity representations with background facts. Secondly, removing the support-specific adjustment module (-adjustment) leads to a significant decline in performance, emphasizing the crucial role of adjusting relation representations to capture meta relational information. Notably, -adjustment suffers the most significant performance degradation on the F-WikiPeople and F-WD50K datasets, which have a large amount of background data. This suggests that the richer the information in the generated relation representations, the more necessary it is to capture meta relational information.

5.5. Analysis on key Parameters

The two key parameters of MetaRH are the maximum number of background facts per entity (L) and relation weight (τ). To analyze the impact of these parameters on MetaRH's performance, experiments are conducted on the largest dataset F-WD50K. Figure 5 illustrates that L significantly affects MetaRH's performance. If L is too small, important background facts may be lost, while too large may result in insufficient attention to the most useful facts. In terms of τ, the optimal performance of MetaRH is achieved when τ is set to 0.9, indicating that relations carry most of the type information of hyper-relational facts, compared to auxiliary attribute-value pairs, in the F-WD50K dataset.

5.6. Case Study

To analyze the practical performance of MetaRH, we randomly selected 6 queries from the F-WD50K dataset for the case study, including 3 hyper-relational facts and 3 binary facts. As shown in Table 4, MetaRH outperforms GANA on most facts,

Query	MetaRH	GANA
((Prince of Wales (Q180729),position held (P39), monarch (Q116)), {(of (P642), Irish Free State (Q31747))})	32	7
((Steve Jobs (Q19837),position held (P39), chief executive officer (Q484876)), {(of (P642), Apple (Q312))})	7	151
((Victor Hugo (Q535),position held (P39), president (Q30461)), {(of (P642), Literary Society (Q3488144))})	81	373
(Second Punic War (Q6271),participant (P710), Macedonia (Q83958))	1	3
(Rhine (Q584),basin country (P205), Switzerland (Q39))	8	8
(Operation Barbarossa (Q83055), participant (P710), Romania (Q203493))	31	44

Table 4: Case study on the F-WD50K dataset. Tail entities in these case facts are assumed to be predicted and are highlighted in red. The second and third columns are the ranks of correct answers for MetaRH and the best baseline GANA, respectively.

demonstrating its superior performance. For the first case of answering the position held by the Prince of Wales in the Irish Free State, MetaRH does not perform as well as GANA. This can be attributed to a lack of background data. Since there are only three facts related to the Irish Free State in the background data, it prevents MetaRH from understanding the auxiliary attribute value pairs.

6. Conclusion

In this paper, we introduced a new task that is practical in real-world scenarios, called Few-Shot Link Prediction on Hyper-relational Facts (FSLPHFs). We defined the task and proposed a solution model called MetaRH, which consists of three modules: relation learning, support-specific adjustment, and query inference. These modules generate initial few-shot relation representations, adjust them based on the support set, and make inferences about queries, respectively. In addition, we constructed three datasets to test our approach. The experimental results show a significant improvement of MetaRH over existing models. In future research, we plan to utilize LLMs to reduce the dependence on background data and training tasks.

7. Acknowledgments

The work is supported by the National Natural Science Foundation of China under grant 62002341, 62306299, the National Key Research and Development Project of China, Beijing Academy of Artificial Intelligence under grant BAAI2019ZD0306, the KGJ Project under grant JCKY2022130C039, SMP-IDATA Open Youth Fund, and the Lenovo-CAS Joint Lab Youth Scientist Project. We thank anonymous reviewers for their insightful comments and suggestions.

8. Bibliographical References

Ralph Abboud, Ismail Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. Boxe: A box embedding model for knowledge base completion. *Advances in Neural Information Processing Systems*, 33:9649–9661.

Ivana Balažević, Carl Allen, and Timothy M Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. *arXiv preprint arXiv:1901.09590*.

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.

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.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. 2019. Meta relational learning for few-shot link prediction in knowledge graphs. *arXiv preprint arXiv:1909.01515*.

Edgar Frank Codd. 1983. A relational model of data for large shared data banks. *Communications of the ACM*, 26(1):64–69.

Shimin Di, Quanming Yao, and Lei Chen. 2021. Searching to sparsify tensor decomposition for n-ary relational data. In *Proceedings of the Web Conference 2021*, pages 4043–4054.

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question answering over freebase with multi-column convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 260–269.

Bahare Fatemi, Perouz Taslakian, David Vazquez, and David Poole. 2019. Knowledge hypergraphs: Extending knowledge graphs beyond binary relations. *arXiv preprint arXiv:1906.00137*.

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.

Mikhail Galkin, Priyansh Trivedi, Gaurav Maheshwari, Ricardo Usbeck, and Jens Lehmann. 2020. Message passing for hyper-relational knowledge graphs. *arXiv preprint arXiv:2009.10847*.

Saiping Guan, Xiaolong Jin, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. 2020. Neuinfer: Knowledge inference on n-ary facts. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 6141–6151.

Saiping Guan, Xiaolong Jin, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. 2021. Link prediction on n-ary relational data based on relatedness evaluation. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):672–685.

Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2018. Shared embedding based neural networks for knowledge graph completion. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 247–256.

Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2019. Link prediction on n-ary relational data. In *The world wide web conference*, pages 583–593.

Zhiyi Jiang, Jianliang Gao, and Xinqi Lv. 2021. Metap: Meta pattern learning for one-shot knowledge graph completion. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2232–2236.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. *Advances in neural information processing systems*, 31.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Hans-Ulrich Krieger and Christian Willms. 2015. Extending owl ontologies by cartesian types to represent n-ary relations in natural language. In *Proceedings of the 1st Workshop on Language and Ontologies*.

Yu Liu, Quanming Yao, and Yong Li. 2020a. Generalizing tensor decomposition for n-ary relational knowledge bases. In *Proceedings of the web conference 2020*, pages 1104–1114.

Yu Liu, Quanming Yao, and Yong Li. 2020b. Generalizing tensor decomposition for n-ary relational knowledge bases. In *Proceedings of the web conference 2020*, pages 1104–1114.

Denis Lukovnikov, Asja Fischer, Jens Lehmann, and Sören Auer. 2017. Neural network-based question answering over knowledge graphs on word and character level. In *Proceedings of the 26th international conference on World Wide Web*, pages 1211–1220.

Haoran Luo, Yuhao Yang, Yikai Guo, Mingzhi Sun, Tianyu Yao, Zichen Tang, Kaiyang Wan, Meina Song, Wei Lin, et al. 2023. Hahe: Hierarchical attention for hyper-relational knowledge graphs in global and local level. *arXiv preprint arXiv:2305.06588*.

Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, GA.

Tsendsuren Munkhdalai and Hong Yu. 2017. Meta networks. In *International conference on machine learning*, pages 2554–2563.

Vinh Nguyen, Olivier Bodenreider, and Amit Sheth. 2014. Don't like rdf reification? making statements about statements using singleton property. In *Proceedings of the 23rd international conference on World wide web*, pages 759–770.

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. In *Proceedings of the 44th International ACM SIGIR conference on research and development in information retrieval*, pages 213–222.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*.

Paolo Rosso, Dingqi Yang, and Philippe Cudré-Mauroux. 2020. Beyond triplets: hyper-relational knowledge graph embedding for link prediction. In *Proceedings of the web conference 2020*, pages 1885–1896.

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. *arXiv preprint arXiv:2010.09638*.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Chenxu Wang, Xin Wang, Zhao Li, Zirui Chen, and Jianxin Li. 2023. Hyconve: A novel embedding model for knowledge hypergraph link prediction with convolutional neural networks. In *Proceedings of the ACM Web Conference 2023*, pages 188–198.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Quan Wang, Haifeng Wang, Yajuan Lyu, and Yong Zhu. 2021. Link prediction on n-ary relational facts: A graph-based approach. *arXiv preprint arXiv:2105.08476*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28.

Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. 2016. On the representation and embedding of knowledge bases beyond binary relations. *arXiv preprint arXiv:1604.08642*.

Bo Xiong, Mojtaba Nayyer, Shirui Pan, and Steffen Staab. 2023. Shrinking embeddings for hyper-relational knowledge graphs. *arXiv preprint arXiv:2306.02199*.

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2018. One-shot relational learning for knowledge graphs. *arXiv preprint arXiv:1808.09040*.

Shiyao Yan, Zequn Zhang, Xian Sun, Guangluan Xu, Shuchao Li, Qing Liu, Nayu Liu, and Shensi Wang. 2022. Polygone: Modeling n-ary relational data as gyro-polygons in hyperbolic space. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4308–4317.

Chuxu Zhang, Huaxiu Yao, Chao Huang, Meng Jiang, Zhenhui Li, and Nitesh V Chawla. 2020.

Few-shot knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*.

Richong Zhang, Junpeng Li, Jiajie Mei, and Yongyi Mao. 2018. Scalable instance reconstruction in knowledge bases via relatedness affiliated embedding. In *Proceedings of the 2018 world wide web conference*, pages 1185–1194.

Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *arXiv preprint arXiv:2305.13168*.