Gaussian Metric Learning for Few-Shot Uncertain Knowledge Graph Completion

Jiatao Zhang, Tianxing Wu, and Guilin Qi(✉)

School of Computer Science and Engineering, Southeast University, Nanjing, China
{zjt,tianxingwu,gqi}@seu.edu.cn

Abstract. Recent advances in relational information extraction have allowed to automatically construct large-scale knowledge graphs (KGs). Nevertheless, an automatic process entails that a significant amount of uncertain facts are introduced into KGs. Uncertain knowledge graphs (UKGs) such as NELL and Probase model this kind of uncertainty as confidence scores associated to facts for providing more precise knowledge descriptions. Existing UKG completion methods require sufficient training examples for each relation. However, most relations only have few facts in real-world UKGs. To solve the above problem, in this paper, we propose a novel method to complete few-shot UKGs based on Gaussian metric learning (GMUC) which could complete missing facts and confidence scores with few examples available. By employing a Gaussian-based encoder and metric function, GMUC could effectively capture uncertain semantic information. Extensive experiments conducted over various datasets with different uncertainty levels demonstrate that our method consistently outperforms baselines.

1 Introduction

Knowledge graphs (KGs) describe structured information of entities and relations, which have been widely used in many intelligent applications such as question-answering and semantic search. Despite large scales of KGs, they are still far from complete to describe infinite real-world facts. In order to complete KGs automatically, many efforts [3,7,18,22,25,31] have been studied to infer missing facts.

Most KGs such as Freebase [2], DBpedia [1], and Wikidata [28] consist of deterministic facts, referred to as Deterministic KGs (DKGs). Due to the automatic process has been widely applied in the construction of large-scale KGs, there are many uncertain facts that make it hard to guarantee the determination of knowledge. Besides, lots of knowledge in some fields such as medicine and finance cannot be represented as deterministic facts. Therefore, Uncertain KGs (UKGs) such as NELL [4] and ConceptNet [24] represent the uncertainty as confidence scores associated to facts. Since such scores could provide more precise information, UKGs benefit many knowledge-driven applications, especially for highly risk-sensitive applications such as drug discovery [23] and investment decisions [19].

C. S. Jensen et al. (Eds.): DASFAA 2021, LNCS 12681, pp. 256–271, 2021.
https://doi.org/10.1007/978-3-030-73194-6_18

Inspired by the completion methods for DKGs, some research efforts [6,13] have been made to complete UKGs. Existing research on UKG completion usually assumes the availability of sufficient training examples for all relations. However, due to the long-tail distribution of relations, most relations only have few facts in real-world UKGs. It is crucial and challenging to deal with such cases.

Table 1. Example facts of relation "synonymfor" and entity "redhat" in NELL.

Relation: synonymfor	Entity: redhat
<(macos, synonymfor, linux), 0.94>	<(redhat, categories, software), 1.00>
<(adobe, synonymfor, flash), 0.94>	<(redhat, categories, enterprise), 1.00>
<(america, synonymfor, us), 1.00>	<(redhat, synonymfor, linux), 1.00>
<(ford, synonymfor, ibm), 1.00>	<(redhat, synonymfor, fedora), 1.00>

In UKGs, entities and relations usually have significant uncertainty of its semantic meaning. For example, in Table 1, the fact *(america, synonymfor, us)* reflects the semantic meaning of *synonymfor* precisely, while other facts such as *(adobe, synonymfor, flash)* and *(macos, synonymfor, linux)* are obviously not precise for the original semantic meaning of *synonymfor*. Another example is entity *redhat*, which has different meaning in facts *(redhat, categories, software)* and *(redhat, categories, enterprise)*. Such a condition is very common in UKGs. We refer this uncertainty as internal uncertainty of entities and relations.

Completing UKGs in few-shot settings is a non-trivial problem for the following reasons: (1) The internal uncertainty of entities and relations is essential to complete UKGs in few-shot scenarios but ignored by previous works. Existing UKG relational learning methods [6,12,13] interpret entities and relations as "points" in low-dimensional spaces. Since there are nothing different about these "points" except their positions, different internal uncertainty of entities and relations cannot be expressed. The ignorance of internal uncertainty leads to insufficient modeling of entities and relations, especially under settings with few and noisy facts. (2) Existing methods of few-shot DKGs completion [30,33] could not be used to complete UKGs directly. These models assume that all facts in KGs are entirely correct without any noise and ignore different qualities of facts. This assumption is obviously not reasonable for UKGs and leads to poor performance in a completion process, which can also be validated in our experiments. Besides, these methods could only complete missing facts but could not estimate confidence scores of completion results.

To address the above issues, we propose a novel method to complete few-shot UKGs based on Gaussian metric learning (GMUC). Given a set of few-shot facts for each relation, our model aims at learning a metric of similarity that can be used to complete missing facts and their confidence scores. Specifically, we first propose a Gaussian neighbor encoder to represent the facts of a relation as multi-dimensional Gaussian distributions, in which the semantic feature (mean) and internal uncertainty (variance) can be learned simultaneously. Gaussian-based representation could innately express internal uncertainty of entities and relations, and enable more expressive parameterization of decision boundaries [26]. Next, a Gaussian matching function considering fact qualities is designed to discover new facts and predict their confidence scores.

In experiments, we newly construct a four datasets under few-shot settings. In order to examine completion performance in real-world UKGs, these datasets have different amounts of noisy facts (i.e., uncertainty levels) to simulate an automatic construction process. We then evaluate our model on two tasks, including link prediction and confidence prediction. The results demonstrate that our model could achieve the best performance on all tasks.

Our main contributions are summarized as follows:

- We are the first to consider the long-tail distribution of relations in UKG completion tasks and formulate the problem as few-shot UKG completion.
- We propose a novel method to complete few-shot UKGs based on Gaussian metric learning. Our model could predict missing facts and their confidence scores by considering fact qualities and internal uncertainty of entities and relations.
- We newly construct a set of datasets for few-shot UKG completion, which contains four datasets with different uncertainty levels.
- We evaluate our model on two tasks, including link prediction and confidence prediction, and our model could achieve promising performances.

2 Related Works

2.1 Completion Methods for DKGs

Various works have been proposed to automatically complete DKGs by learning relation representation. RESCAL [18] represents inherent structures of relational data as tensor factorization. TransE [3] regards the relation between entities as a translation operation on low-dimensional embeddings. More advanced approaches have been invested, such as DistMult [31] and ComplEx [25]. Recently, methods utilizing deep neural networks, such as ConvE [7], have also been proposed.

2.2 Completion Methods for UKGs

Inspired by completion methods for DKGs, some UKG completion methods are have also been invested. UKGE [6] is the first UKG embedding model that is able to capture both semantic and uncertain information in embedding space. GTransE [13] uses confidence-margin-based loss function to deal with uncertainty on UKGs. PKGE in [12] employs Markov Logic Network (MLN) to learn first-order logic and encodes uncertainty.

2.3 Few-Shot Learning

Recent few-shot learning methods can be divided into two categories: (1) Metric-based methods [15,21,27,32], trying to learn a similarity metric between new instances and instances in the training set. Most of the methods in this category use the general matching framework proposed in the deep siamese networks given in [15]. An example is the matching networks [27], which make predictions by comparing input examples with small support set with labels. (2) Methods based on meta-learners [9,16,17,20], aiming to directly predict or update parameters of the model according to training data.

Recently, few-shot learning has been applied in DKG completion. Gmatching [30] designs a matching metric by considering both learned embeddings and local-subgraph structures. FSRL [33] proposes a more effective neighbor encoder module to capture heterogeneous graph structure of knowledge. Unlike metric-based methods, MetaR [5] focuses on transferring meta information of a relation to learn models effectively and efficiently. CogKR [8] solves one-shot DKG completion problem by combining summary and reasoning modules. However, these methods are intractable for UKG completion since they ignore the fact qualities and cannot predict confidence scores.

As far as we know, this is the first work to study the few-shot UKG completion problem.

3 Problem Definition

Definition 1. Uncertain Knowledge Graph
An Uncertain Knowledge Graph (UKG) is denoted as $G = \{< (h, r, t), s >\}$, where $(h, r, t) \in E \times R \times E$ represents a fact as a triple, E and R are the sets of all entities and relations, $s \in [0, 1]$ is the confidence score which means the confidence of this triple to be true.

Definition 2. Few-Shot Uncertain Knowledge Graph Completion
For a relation r and one of its head entities h_j in an UKG G, few-shot UKG problem is to predict corresponding tail entities and confidence scores based on a few-shot support set $\mathcal{S}_r = \{< (h_i, t_i), s_i > |(< (h_i, r, t_i), s_i > \in G\}$. The problem can be formally represented as $r :< (h_j, ?), ? >$.

Table 2. Examples of a training task and a testing task in 3-shot UKG completion problem

Phase	Training	Testing
Task	Relation: productby	Relation: synonymfor
Support set	<(word, microsoft), 1.00>	<(us, america), 1.00>
	<(alphago, google), 0.50>	<(adobe, flash), 0.94>
	<(ps4, nintendo), 0.37>	<(linux, microsoft), 0.50>
Query	<(iphone, apple), 0.94>	(Hewlett-Packard, HP)

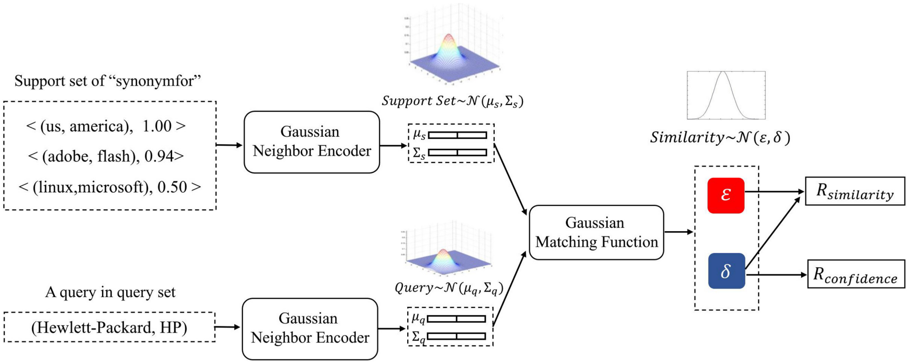


Fig. 1. The framework of GMUC: it first encodes support set and queries into multi-dimensional Gaussian distributions by Gaussian neighbor encoders. Then a Gaussian matching function is employed to construct similarity distribution between queries and the support set which is denoted as *Similarity*. Based on *Similarity*, we define two matching results $R_{similarity}$ and $R_{confidence}$ to complete missing triples and confidence scores.

As above definition, a few-shot UKG completion task can be always defined for a specific relation. During a testing process, there usually is more than one triple to be completed. We denote such triples as a query set $\mathcal{Q}_r = \{r :< (h_j, ?)\, , ? >\}$ (Table 2).

A few-shot UKG completion method aims to gain the capability to predict new triples and their confidence scores about a relation r with only observing a few triples about r. Therefore, its training process is based on a set of tasks $\mathcal{T}_{train} = \{\mathcal{T}_i\}_{i=1}^{M}$, where $\mathcal{T}_i = \{\mathcal{S}_i, \mathcal{Q}_i\}$ indicates an individual few-shot UKG completion task. Every task has its own support set and query set. In the testing process, a set of new tasks $\mathcal{T}_{test} = \{\mathcal{T}_j\}_{j=1}^{N}$ ($\mathcal{T}_{test} \cap \mathcal{T}_{train} = \emptyset$), which can be constructed similarity.

4 Methodology

In this section, we present the detail of our proposed model GMUC (Fig. 1). First, we give the architecture of Gaussian neighbor encoder to represent queries and support set by capturing the semantic information and uncertainty. Then, we focus on Gaussian matching to measure the similarity between the queries and the support set. Finally, we describe the learning process of GMUC.

4.1 Gaussian Neighbor Encoder

Many point-based UKG relational learning methods [6,12,13] have desirable performances with sufficient training data, but these models ignore internal uncertainty which is essential in few-shot settings. Inspired by [10,33], we design the

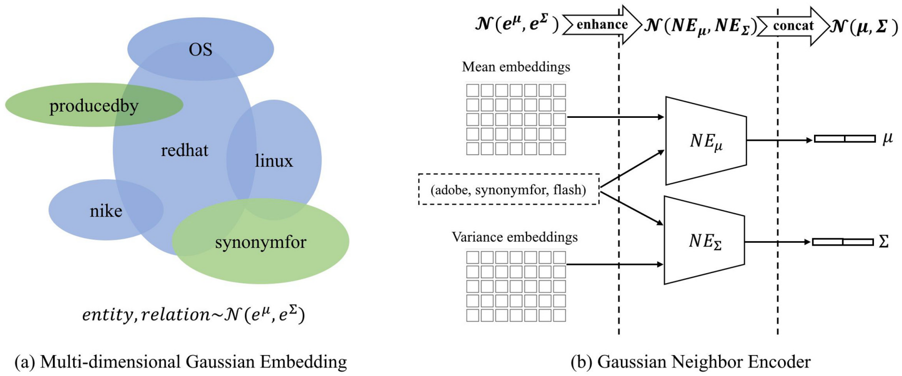


Fig. 2. (a) Multi-dimensional Gaussian embedding for entities and relations. The mean embeddings e^{μ} of such multi-dimensional Gaussian distribution indicates its semantic feature, and the variance embeddings e^{Σ} indicates the corresponding internal uncertainty; (b) A diagram of Gaussian Neighbor Encoder. Two neighbor encoders NE_{μ} and NE_{Σ} are employed to learn enhanced mean embeddings μ and variance embeddings Σ of triples respectively.

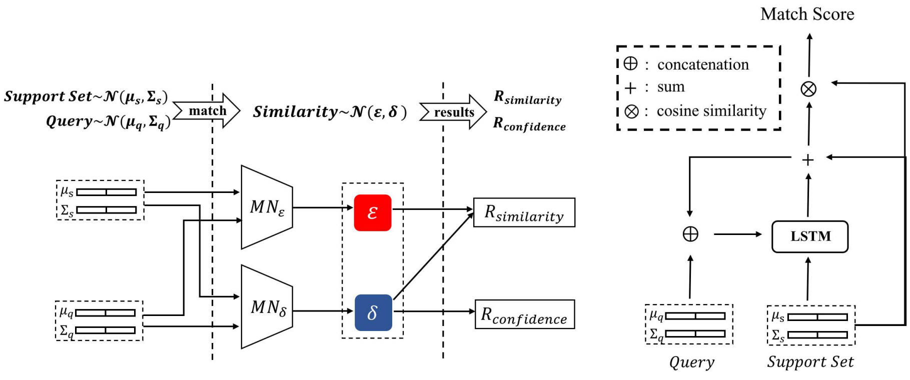
Gaussian neighbor encoder to encode a support set and queries, which could naturally capture internal uncertainty by employing multi-dimensional distributions.

As Fig. 2(a) shows, we first represent each entity and relation as a multi-dimensional Gaussian distribution $\mathcal{N}(e^{\mu}, e^{\Sigma})$, where $e^{\mu} \in \mathbb{R}^{d\times 1}$ is the mean embedding and $e^{\Sigma} \in \mathbb{R}^{d\times 1}$ is the variance embedding of entity or relation, d is embedding dimension. The mean embedding indicates its semantic feature, and the variance embedding indicates the corresponding internal uncertainty.

Based on the Gaussian-based representation of entities and relations, we then use heterogeneous neighbor encoders [33] to enhance the representation of each entity with its local structure in a knowledge graph (Fig. 2(b)). Specifically, for a entity h, we denote the enhanced representation as $\mathcal{N}(NE_{\mu}(h), NE_{\Sigma}(h))$, where NE_{μ} and NE_{Σ} are two heterogeneous neighbor encoders. The set of neighbors of h denoted as $N_h = \{<(r_i, t_i), s_i> | <(h, r_i, t_i), s_i> \in G\}$, where r_i and t_i represent the i-th relation and corresponding tail entity of h, s_i is confidence score of this triple. Besides, an attention module is introduced to consider different impacts of neighbors $<(r_i, t_i), s_i> \in N_h$. The calculation process of a heterogeneous neighbor encoder is defined as follows:

$$NE_{*}(h) = Tanh(\sum_i s_i \alpha_i e^{*}_{t_i}) \tag{1}$$

$$\alpha_i = \frac{\exp\left\{u_{rt}^{T}\left(W_{rt}\left(e^{*}_{r_i} \oplus e^{*}_{t_i}\right) + b_{rt}\right)\right\}}{\sum_j \exp\left\{u_{rt}^{T}\left(W_{rt}\left(e^{*}_{r_j} \oplus e^{*}_{t_j}\right) + b_{rt}\right)\right\}} \tag{2}$$



(a) Gaussian Matching Function

(b) Matching Network

Fig. 3. (a) A diagram of Gaussian matching function. Two LSTM-based matching networks MN_ε and MN_δ are used to calculate mean values ε and variance values δ of *Similarity*, respectively. (b) The structure of a LSTM-based matching network.

where $*$ could be μ or Σ, $e^{\mu}_{t_i}$ and $e^{\mu}_{r_i}$ are mean embeddings of t_i and r_i, $e^{\Sigma}_{t_i}$ and $e^{\Sigma}_{r_i}$ are variance embeddings of t_i and r_i. Moreover, $u_{rt} \in \mathbb{R}^{d\times 1}$, $W_{rt} \in \mathbb{R}^{d\times 2d}$ and $b_{rt} \in \mathbb{R}^{d\times 1}$ are learnable parameters, $\oplus$ is a concatenation operator.

Each triple in a support set and queries is interpreted as $\mathcal{N}(\mu, \Sigma)$, where mean embedding μ and variance embedding Σ are defined as follows:

$$\mu = [NE_\mu\,(h_k) \oplus NE_\mu\,(t_k)] \tag{3}$$

$$\Sigma = [NE_\Sigma\,(h_k) \oplus NE_\Sigma\,(t_k)] \tag{4}$$

By above approach, we can get the representation of a query $\mathcal{N}\,(\mu_q, \Sigma_q)$. For the representations of support set $\{\mathcal{N}\,(\mu_i, \Sigma_i)\,|{<}(h_i, t_i)\,, s_i{>} \in \mathcal{S}_r\}$, we use max-pooling to aggregate these distributions into one multi-Gaussian distribution $\mathcal{N}\,(\mu_s, \Sigma_s)$, where μ_s and Σ_s are defined as follows:

$$\mu_s = pool_{max}(s_i \cdot \mu_i) \tag{5}$$

$$\Sigma_s = pool_{max}(s_i \cdot \Sigma_i) \tag{6}$$

4.2 Gaussian Matching Function

Given the Gaussian neighbor encoder module, we now present the Gaussian matching function to measure the similarity of queries and the support set (Fig. 3(a)). Most existing metric-based functions complete missing triples by a single value similarity, but they cannot give confidence scores to the completion results, which is inadequate for UKG completion. To address this issue, we propose Gaussian matching function to complete missing triples and their confidence scores simultaneously.

We first define the matching similarity *Similarity* as a one-dimensional Gaussian distribution $\mathcal{N}(\varepsilon, \delta)$, where mean value $\varepsilon \in \mathbb{R}$ can be regarded as the most likely similarity value and the variance value $\delta \in [0, 1]$ refers to the uncertainty of such similarity value.

Then, we employ LSTM-based matching networks [27] to calculate *Similarity*. Compared with a simple cosine similarity, the matching networks perform a multi-step matching process, which could effectively improve the matching capability [30]. Figure 3(b) shows the structure of a matching network. The calculation process of a matching network *MN* is defined as follows:

$$\begin{aligned} &MN(x, y) = g_t \cdot x \\ &g_t = g_t' + y \\ &g_t', c_t = LSTM(y, [g_{t-1} \oplus x, c_{t-1}]) \end{aligned} \tag{7}$$

where x and y are embeddings to be matched, $LSTM(z, [g_t, c_t])$ is a LSTM cell [11] with input z, hidden state g_t and cell state c_t. After T processing steps, we use the inner product between g_t and x as the matching score of x and y.

Two matching networks MN_ε and MN_δ are used to get the mean value ε and variance value δ of *Similarity* by the following formulas:

$$\varepsilon = MN_\varepsilon(\mu_s, \mu_q) \tag{8}$$

$$\delta = sigmoid(W \cdot MN_\delta(\Sigma_s, \Sigma_q) + b) \tag{9}$$

where $sigmoid(x) = 1/(1 + exp(-x))$, W and b are learnable parameters.

To complete missing triples and their confidence scores, we define two matching results $R_{similarity}$ and $R_{confidence}$ based on the *Similarity* as follows:

$$R_{similarity} = \varepsilon + \lambda(1 - \delta) \tag{10}$$

$$R_{confidence} = 1 - \delta \tag{11}$$

where λ is hyper-parameter. Finally, we use the $R_{similarity}$ as ranking scores to complete missing triples and the $R_{confidence}$ to predict confidence scores.

4.3 The Learning Process

For a relation r, we randomly sample a set of few positive entity pairs $\{< (h_k, t_k), s_k > \mid < (h_k, r, t_k), s_k > \in G\}$ and regard them as the support set $\mathcal{S}_r$. The remaining positive entity pairs $\mathcal{Q}_r = \{< (h_l, t_l), s_l > \mid < (h_l, r, t_l), s_l > \in G \cap < (h_l, t_l), s_l > \notin \mathcal{S}_r\}$ are utilized as positive queries.

$\mathcal{L}_{mse}$ is designed to minimize the mean squared error (MSE) between the ground truth confidence score s and our predicting confidence score $R_{confidence}$ for each triple $< (h, t), s > \in \mathcal{Q}_r$. Specifically, $\mathcal{L}_{mse}$ is defined as:

$$\mathcal{L}_{mse} = \sum_{<(h,t),s>\in\mathcal{Q}_r} |R_{confidence} - s|^2 \tag{12}$$

Following TransE [3], we design a margin-based ranking loss $\mathcal{L}_{rank}$ to make the mean value ε of positive entity pairs to be higher than those of negative entity pairs. In order to reduce the impact of poor quality queries, we filter queries by threshold thr and get $\mathcal{Q}_r^{thr} = \{< (h_l, t_l), s_l > \mid < (h_l, r, t_l), s_l > \in \mathcal{Q}_r \ and \ s_l \geq thr\}$. Then we construct a group of negative entity pairs $\mathcal{Q}_r^{thr-} = \left\{ \left(h_l, t_l^- \right) \middle| < \left(h_l, r, t_l^- \right), * > \notin G \right\}$ by polluting the tail entities. The ranking loss is formulated as:

$$\mathcal{L}_{rank} = \sum_{<(h,t),s> \in \mathcal{Q}_r^{thr}} \sum_{(h,t') \in \mathcal{Q}_r^{thr-}} s \cdot [\gamma + \varepsilon_{(h,t)} - \varepsilon_{(h,t')}]_+ \tag{13}$$

where $[x]_+ = max[0, x]$ and γ is a safety margin distance, $\varepsilon_{(h,t)}$ and $\varepsilon_{(h,t')}$ are mean value of $Similarity$ between query $(h, t_l/t_l')$ and support set $\mathcal{S}_r$. Here the triple confidence score s instructs our model to pay more attention on those more convincing queries.

Finally, we define the final objective function as:

$$\mathcal{L}_{joint} = \mathcal{L}_{rank} + \mathcal{L}_{mse} \tag{14}$$

Our objective is to minimize $\mathcal{L}_{joint}$ in the training process for all query tasks. The detail of this process can be summarized in Algorithm 1.

Algorithm 1: GMUC Training Procedure

Input:
a) Meta-training task (relation) set $\mathcal{T}_{train}$;
b) Embeddings of entities and relations φ;
c) Initial parameters θ of the metric model

for *epoch:=0 to* MAX_{epoch} **do**
  for $\mathcal{T}_r$ *in* $\mathcal{T}_{train}$ **do**
    Sample few entity pairs as support set $\mathcal{S}_r$
    Sample a batch of positive queries $\mathcal{Q}_r$ and filtered queries $\mathcal{Q}_r^{thr}$
    Pollute the tail entity of queries to get $\mathcal{Q}_r^{thr-}$
    Calculate the loss by Eq. (14)
    Update parameters θ and φ
return Optimal model parameters θ and φ

5 Experiments

In this section, we present the detail of experiments. First, we introduce newly constructed datasets under few-shot settings. Then, we describe baseline models and the experimental setup. Finally, we evaluate our model on two tasks, including link prediction and confidence prediction.

5.1 Datasets

In this paper, we evaluated our model based on NL27K [6], which is a typical UKG dataset extracted from NELL [4]. However, the triples in NL27K are high quality (confidence scores $\geq$0.95) which rarely has noises or uncertain data. Therefore, similar to the work given in CRKL [29], we generated new datasets with different amounts of noisy triples (i.e., uncertainty levels) based on NL27K to simulate the real-world UKGs constructed by an automatic process with less human supervision. Specifically, based on NL27K, we constructed four datasets: NL27K-N0, NL27K-N1, NL27K-N2 and NL27K-N3 which include 0%, 10%, 20% and 40% negative triples of positive triples. Then we utilized CKRL [29] to assign confidence scores to the triples in datasets. The confidence scores are calculated by the following function:

$$C(h,r,t) \ = \omega_1 \cdot LT(h,r,t) + \omega_2 \cdot PP(h,r,t) + \ \omega_3 \cdot AP(h,r,t) \tag{15}$$

where LT is the local triple confidence which concentrates on the inside of a triple. PP is the prior path confidence which utilizes the co-occurrence of a relation and a path to represent their dissimilarity. AP is the adaptive path confidence which could flexibly learn relation-path qualities. ω_1, ω_2 and ω_2 are hyper-parameters. Following [29], we selected $\omega_1 = 0.75$, $\omega_2 = 0.05$ and $\omega_3 = 0.2$ to create the datasets.

After assigning confidence scores to triples of datasets, following [30], we selected the relations with less than 500 but more than 50 triples to construct few-shot tasks. We referred to the rest of the relations as background relations since their triples provide important background knowledge to match entity pairs. Table 3 shows the datasets statistics. We used 101/13/20 tasks for training/validation/testing separately.

Table 3. Statistics of the Datasets. #Entities denotes the number of unique entities and #Relations denotes the number of all relations. #Triples denotes the number of all triples. #Tasks denotes the number of relations we used as few-shot tasks. #Neg_Triples denotes the number of negative examples. Avg(s) and Std(s) are the average and standard deviation of the confidence scores.

Dataset	#Entities	#Relations	#Triples	#Tasks
NL27K-N0	27,221	404	175,412	134
Datasets	NL27K-N0	NL27K-N1	NL27K-N2	NL27K-N3
#Neg_Triples	0	17,541	35,082	70,164
Avg(s)	0.863	0.821	0.787	0.732
Std(s)	0.111	0.176	0.210	0.244

5.2 Baseline Methods

Three categories of baseline methods are considered.

Embedding Models for UKG Completion. UKGE [6] is a recently proposed UKG embedding model. UKGE preserves the semantic and uncertain information by matching the representation of entities and relations in embedding space.

Metric-based Models for Few-Shot DKG Completion. GMatching [30] and FSRL [33] are metric-based few-shot DKG completion models. However, these models cannot deal with the confidence scores of triples, which may suffer poor-quality triples. Besides, these models can only complete missing triples but cannot predict their confidence scores.

Variant Models of GMUC. We proposed two variant models of GMUC, called GMUC-noconf and GMUC-point. GMUC-noconf removes all the processes considering triple qualities. GMUC-point only uses the mean embedding μ of queries and a support set and the mean value ε of *Similarity* to calculate ranking scores and confidence scores, which is a point-based model.

5.3 Experimental Setup

Adam optimizer [14] is used for training. For baseline models, we reported results based on their best hyper-parameter. We identified each model based on the validation set performance. For hyper-parameter tuning, we searched the best hyper-parameter as follows: learning rate $lr \in \{0.001, 0.005, 0.01\}$, dimension $d \in \{64, 128, 256, 512\}$, batch size $b \in \{128, 256, 512, 1024\}$, margin $\gamma \in \{1.0, 5.0\}$, threshold $thr \in \{0.2, 0.3, 0.4, 0.5\}$, trade-off factor $\lambda \in \{0.1, 0.5, 1.0\}$. The training was stopped using early stopping based on Hit@10 on the validation set, computed every 10 epochs. The maximum number of local neighbors in Gaussian neighbor encoder is set to 30 for all datasets. For the LSTM module in Gaussian matching function, the hidden state is set to 128 and the number of recurrent steps equals 2. Specifically, for each dataset, the optimal configuration is $\{lr = 0.01, d = 128, b = 128, \gamma = 5.0\}$. For NL27K-N0, NL27K-N1 and NL27K-N2, we set $\{thr = 0.3, \lambda = 0.1\}$, while $\{thr = 0.5, \lambda = 1.0\}$ for NL27K-N3. Besides, the few-shot size $|\mathcal{S}_r|$ is set to 3 for the following experiments.

5.4 Link Prediction

This task is to complete missing tail entities for a given relation r and a head entity h,denoted as $(h, r, ?)$.

Evaluation Protocol. We followed the same protocol as in FSRL [33]: In the testing phase, for each positive query <(h, r, t), s>, we replaced the tail entity by candidate entities in UKG and ranked these entities in descending order of ranking scores. The maximum candidate entities size is set to 1000 for all datasets. Based on these entity ranking lists, we used three evaluation metrics by aggregation over all the queries: first, the mean reciprocal rank of correct entities (denoted as MRR); then, the proportion of correct entities in the top1 and top10 in entity ranking lists (denoted as Hits@1, Hits@10). A good method should obtain higher MRR, Hits@1 and Hits@10. Considering some corrupted triples for (h, r, t) also exists in datasets, such a prediction should also be regarded correct. To eliminate this factor, we removed those corrupted triples that already appear in training, validation and testing sets before obtaining the ranking entity list of each query. We termed the setting as "Filter" and used it for our evaluation.

Table 4. Result of link prediction

Dataset	NL27K-N0			NL27K-N1			NL27K-N2			NL27K-N3		
Metrics	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10
GMatching	0.361	0.272	0.531	0.193	0.123	0.315	0.125	0.066	0.253	0.025	0.005	0.051
FSRL	0.397	0.304	0.589	0.188	0.101	0.333	0.123	0.052	0.264	0.027	0.007	0.045
UKGE	0.053	0.058	0.138	0.071	0.107	0.153	0.057	0.066	0.153	0.092	0.091	0.144
GMUC-noconf	0.420	0.324	0.611	0.179	0.113	0.310	0.127	0.071	0.271	0.092	0.048	0.155
GMUC-point	0.413	0.316	0.603	0.215	0.130	**0.344**	0.131	**0.113**	0.272	0.065	0.006	0.156
GMUC	**0.433**	**0.342**	**0.644**	**0.219**	**0.148**	0.332	**0.143**	0.110	**0.292**	**0.148**	**0.107**	**0.194**

Results. Table 4 shows the results of link prediction in datasets with different uncertainty levels, from which we could observe that:

(1) Our model outperforms baselines on all datasets. Compared with UKGE, GMUC has consistent improvements, demonstrating that GMUC could better complete a UKG in few-shot settings. Additionally, GMUC outperforms few-shot DKG completion methods (i.e., GMatching and FSRL), especially for NL27K-N3 GMUC achieves 0.194 of Hit@10 while GMatching only has 0.051, which indicates the promising effectiveness of GMUC for KG completion in uncertain scenarios.
(2) Comparing evaluation results between different datasets, we found that FSRL and Gmatching achieve good performance for NL27K-N0 but have a great descent when the uncertainty level goes up. Taking Hit@10 of FSRL as an example, it achieves 0.589 for NL27K-N0, but it only has 0.045 for NL27K-N3. It demonstrates the few-shot DKG completion methods could not be used to complete UKGs directly. Conversely, the performance of UKGE is worse than FSRL and GMatching for the datasets with lower uncertainty level, including NL27K-N0, NL27K-N1 and NL27K-N3, but keeps stable from NL27K-N0 to NL27K-N3. GMUC consistently outperforms FSRL, GMatching and UKGE. A possible reason is that FSRL and GMatching are based on ranking loss which is sensitive for noisy data, while UKGE is based on MSE-loss which is better suitable for noisy data in UKGs. GMUC based on the similarity distribution which can be regarded as a combination of these two methods.

(3) For a more detailed analysis of the component effectiveness, we compared GMUC and its variant models to do ablation studies. First, to investigate the design of the multi-dimensional Gaussian representation of Gaussian neighbor encoder, we compared GMUC with GMUC-point which can be seen as a point-based function. The results of GMUC-point are worse than GMUC, demonstrating the benefit of Gaussian representation. It can also suggest that confidence scores of triples are essential information that could be used to enhance the performance of link prediction. Then, GMUC outperforms GMUC-noconf, demonstrating our strategy considering the triple qualities is useful.

5.5 Confidence Prediction

The objective of this task is to predict confidence scores of triples, formulated as $<(h,r,t),?>$.

Evaluation Protocol. For each triple (h,r,t) in the query set, we predicted its confidence score and reported the Mean Squared Error (MSE) and Mean Absolute Error (MAE).

Table 5. Result of confidence prediction

Dataset	NL27K-N0		NL27K-N1		NL27K-N2		NL27K-N3	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
UKGE	0.070	0.198	0.061	0.177	0.063	0.184	0.072	0.199
GMUC-noconf	0.019	0.106	0.022	0.111	0.028	0.126	0.029	0.130
GMUC-point	0.038	0.154	0.035	0.143	0.042	0.156	0.046	0.157
GMUC	**0.013**	**0.086**	**0.018**	**0.096**	**0.022**	**0.104**	**0.027**	**0.113**

Results. Table 5 shows the results of confidence prediction. We could find that:

(1) UKGE has larger MAE and MSE for NL27K-N0 (few-shot settings dataset) than the original NL27K (non-few-shot dataset) in [33], which validates that UKGE could not complete UKGs well in few-shot scenarios. Our model consistently outperforms UKGE, demonstrating the effectiveness of GMUC for UKG completion in few-shot settings. Comparing with evaluation results between different datasets, we found that our model and UKGE keep stable. It is the reason why these methods can get stable results on the link prediction task with different uncertainty levels.
(2) To investigate the effect of using Gaussian-based representation, we compared GMUC and its variant model GMUC-point which could be regarded as a point-based method. The results of GMUC-point are worse than

GMUC, demonstrating the benefit of Gaussian representation. A possible reason why the Gaussian representation could enhance the completion performance is that the point-based UKG completion methods try to capture the semantic and uncertain information in one embedding space simultaneously, while Gaussian representation uses mean embedding and variance embedding to learn such information respectively in two embedding spaces with different learning targets.

(3) By comparing GMUC and GMUC-noconf, we could find that our strategy considering triple qualities can also improve the performance of confidence prediction.

6 Conclusion and Future Work

In this paper, we proposed a novel method to complete few-shot UKGs based on Gaussian metric learning (GMUC), which could complete missing triples and confidence scores with few examples available. Compared with the state-of-the-art UKG completion model and few-shot DKG completion models on few-shot UKG datasets, our model has comparable effectiveness of capturing uncertain and semantic information. Experimental results also show our method consistently outperforms baselines in datasets with different uncertain levels. The source code and datasets of this paper can be obtained from https://github.com/zhangjiatao/GMUC.

In the future, we will explore the following research directions:

(1) The meta-information of relation could provide the common knowledge which could help model learn more efficiently in few-shot setting. We will explore to combine the metric-based method and meta-based method to better complete UKG with few examples.
(2) We observe that the uncertainty levels of datasets could highly adverse the completion result in few-shot settings. In the future, we may design a metric to measure the uncertainty of data, which is set by manual in this work.
(3) External knowledge such as logic rules could enrich KGs. Our future work will introduce external knowledge to further enhance the precision of completion.

Acknowledgements. This work was partially supported by the National Key Research and Development Program of China under grants (2018YFC0830200, 2017YFB1002801), the National Natural Science Foundation of China grants (U1736204, 62006040), and the Judicial Big Data Research Centre, School of Law at Southeast University. In addition, we wish to thank Prof. Qiu Ji for her valuable suggestion.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC-2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52
2. Bollacker, K.D., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, pp. 1247–1250 (2008)
3. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of the 2013 Annual Conference on Neural Information Processing Systems, NeurIPS 2013, pp. 2787–2795 (2013)
4. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr., E.R.H., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: Proceedings of the 2010 AAAI Conference on Artificial Intelligence, AAAI 2010 (2010)
5. Chen, M., Zhang, W., Zhang, W., Chen, Q., Chen, H.: Meta relational learning for few-shot link prediction in knowledge graphs. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, EMNLP 2019, pp. 4216–4225 (2019)
6. Chen, X., Chen, M., Shi, W., Sun, Y., Zaniolo, C.: Embedding uncertain knowledge graphs. In: Proceedings of the 2019 AAAI Conference on Artificial Intelligence, AAAI 2019, pp. 3363–3370 (2019)
7. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: Proceedings of the 2018 AAAI Conference on Artificial Intelligence, AAAI 2018, pp. 1811–1818 (2018)
8. Du, Z., Zhou, C., Ding, M., Yang, H., Tang, J.: Cognitive knowledge graph reasoning for one-shot relational learning. CoRR (2019)
9. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 2017 International Conference on Machine Learning, ICML 2017, vol. 70, pp. 1126–1135 (2017)
10. He, S., Liu, K., Ji, G., Zhao, J.: Learning to represent knowledge graphs with Gaussian embedding. In: Proceedings of the 2015 ACM International Conference on Information and Knowledge Management, CIKM 2015, pp. 623–632 (2015)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–1780 (1997)
12. Huang, Z., Iyer, R.G., Xiao, Z.: Uncertain knowledge graph embedding using probabilistic logic neural networks (2017)
13. Kertkeidkachorn, N., Liu, X., Ichise, R.: GTransE: generalizing translation-based model on uncertain knowledge graph embedding. In: Ohsawa, Y., et al. (eds.) JSAI 2019. AISC, vol. 1128, pp. 170–178. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-39878-1_16
14. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) Proceedings of the 2015 International Conference on Learning Representations, ICLR 2015 (2015)
15. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: Proceedings of the 2015 Workshop on International Conference on Machine Learning, ICML 2015, vol. 2 (2015)

16. Li, Z., Zhou, F., Chen, F., Li, H.: Meta-SGD: Learning to learn quickly for few shot learning. CoRR abs/1707.09835 (2017)
17. Munkhdalai, T., Yu, H.: Meta networks. Proc. Mach. Learn. Res. **70**, 2554 (2017)
18. Nickel, M., Tresp, V., Kriegel, H.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 2011 International Conference on Machine Learning, ICML 2011, pp. 809–816 (2011)
19. Qi, G., Gao, H., Wu, T.: The research advances of knowledge graph. Technol. Intell. Eng. **3**, 4–25 (2017)
20. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: Proceedings of the 2017 International Conference on Learning Representations, ICLR 2017 (2017)
21. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. In: Proceedings of the 2017 Annual Conference on Neural Information Processing Systems, NeurIPS 2017, pp. 4077–4087 (2017)
22. Socher, R., Chen, D., Manning, C.D., Ng, A.Y.: Reasoning with neural tensor networks for knowledge base completion. In: Proceedings of the 2013 Annual Conference on Neural Information Processing Systems, NeurIPS 2013, pp. 926–934 (2013)
23. Sosa, D.N., Derry, A., Guo, M., Wei, E., Brinton, C., Altman, R.B.: A literature-based knowledge graph embedding method for identifying drug repurposing opportunities in rare diseases. bioRxiv, p. 727925 (2019)
24. Speer, R., Chin, J., Havasi, C.: ConceptNet 5.5: an open multilingual graph of general knowledge. In: Proceedings of the 2017 AAAI Conference on Artificial Intelligence, AAAI 2017, pp. 4444–4451 (2017)
25. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of the 2016 International Conference on Machine Learning, ICML 2016, pp. 2071–2080 (2016)
26. Vilnis, L., McCallum, A.: Word representations via gaussian embedding. In: Bengio, Y., LeCun, Y. (eds.) Proceedings of the 2015 International Conference on Learning Representations, ICLR 2015 (2015)
27. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching networks for one shot learning. In: Proceedings of the 2016 Annual Conference on Neural Information Processing Systems, NeurIPS 2016, pp. 3630–3638 (2016)
28. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**, 78–85 (2014)
29. Xie, R., Liu, Z., Lin, F., Lin, L.: Does William Shakespeare REALLY Write Hamlet? knowledge representation learning with confidence. In: Proceedings of the 2018 AAAI Conference on Artificial Intelligence, AAAI 2018, pp. 4954–4961 (2018)
30. Xiong, W., Yu, M., Chang, S., Wang, W.Y., Guo, X.: One-shot relational learning for knowledge graphs. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2018, pp. 1980–1990 (2018)
31. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of the 2015 International Conference on Learning Representations, ICLR 2015 (2015)
32. Yu, M., et al.: Diverse few-shot text classification with multiple metrics. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, pp. 1206–1215 (2018)
33. Zhang, C., Yao, H., Huang, C., Jiang, M., Li, Z., Chawla, N.V.: Few-shot knowledge graph completion. In: Proceedings of the 2020 AAAI Conference on Artificial Intelligence, AAAI 2020, pp. 3041–3048 (2020)