Incorporating Uncertainty of Entities and Relations into Few-Shot Uncertain Knowledge Graph Embedding

Jingting Wang, Tianxing Wu(✉), and Jiatao Zhang

Southeast University, Nanjing, China
{jtwang,tianxingwu,zjt}@seu.edu.cn

Abstract. In this paper, we study the problem of embedding few-shot uncertain knowledge graphs. Observing the existing embedding methods may discard the uncertainty information, or require sufficient training data for each relation, we propose a novel method by incorporating the inherent uncertainty of entities and relations (i.e. element-level uncertainty) into uncertain knowledge graph embedding. We introduce different metrics to quantify the uncertainty of different entities and relations. By employing a metric-based framework, our method can effectively capture both semantic and uncertainty information of entities and relations in the few-shot scenario. Experimental results show that our proposed method can learn better embeddings in terms of the higher accuracy in both confidence score prediction and tail entity prediction.

Keywords: Uncertain knowledge graph · Knowledge graph embedding · Few-shot learning

1 Introduction

Knowledge graphs (KGs) [1] describe the real-world facts in the form of triples *(head entity, relation, tail entity)*, indicating that two entities are connected by a sepecific relation. In addition to deterministic KGs (DKGs), much recent attention has been paid to uncertain KGs (UKGs). UKGs, such as Probase [2], NELL [3] and ConceptNet [4], associate each fact (or triple) with a confidence score representing the likelihood of that fact to be true, e.g. *(Twitter, competeswith, Facebook, 0.85)*. Such uncertain knowledge representations can capture the uncertain nature of reality, and provide more precise reasoning.

KG embedding models are essential tools for incorporating the structured knowledge representations in KGs into machine learning. These models encode entities and relations into continuous vector spaces, so as to accurately capture the similarity of entities and preserve the structure of KGs in the embedding space. Inspired by the works about DKG embeddings [5–8], some efforts have been devoted to UKGs [9–12] embedding. Existing methods usually assume the availability of sufficient training examples for all relations. However, the frequency distributions of relations in real datasets often have long tails, which

M. Sun et al. (Eds.): CCKS 2022, CCIS 1669, pp. 16–28, 2022.
https://doi.org/10.1007/978-981-19-7596-7_2

means that a large potion of relations appear in only a few triples in KGs. It is important and challenging to deal with the relations with limited number of triples, leading to the few-shot UKG embedding problem.

To our knowledge, GMUC [13] is the first and the only embedding method designed for few-shot UKGs. GMUC represents each entity and relation as a multi-dimensional Gaussian distribution, and utilizes a metric-based framework to learn a matching function for completing missing facts and their confidence scores. In light of the success of GMUC, we further find an important issue to be solved. Although the variance vectors in Gaussian distribution are claimed to represent entities' or relations' uncertainty, it lacks rational explanations and how such a setting influences the process of UKG embedding is unpredictable, which may lead to the imprecise modeling of entities and relations.

To alleviate this problem, we consider the uncertainty of an entity/relation on its semantic level. Table 1 shows an example in NELL. For relation *museumincity*, the head entity must be a museum, and the corresponding tail entity must be a city. In contrast, relation *atlocation* provides more rich semantics because it can represent the connection between company and city, country and continent, or person and country, etc. Obviously, *atlocation* has higher uncertainty than *museumincity*. Similarly, different entities has different uncertainty extents. For example, *Alice* is more certain than *artist*, since there are much more persons belong to *artist* category. To summarize, the uncertainty of an entity/relation can be measured by its semantic imprecision. In contrast to triple-level uncertainty (i.e., the confidence score of a triple), we call this kind of uncertainty as *element-level* uncertainty of UKGs.

Table 1. Example facts of relations *museumincity* and *atlocation* in NELL.

Relation: museumincity	Relation: atlocation
(Gotoh Museum, Tokyo, 1.0)	(Air Canada, Vancouver, 0.92)
(Decordova, Lincoln, 0.96)	(Albania, Europe, 1.0)
(Whitney Museum, New York, 0.93)	(Queen Victoria, Great Britain, 0.93)

In this paper, we propose a new few-shot UKG embedding model by incorporating the inherent uncertainty of entities and relations. In order to capture the element-level uncertainty in UKGs, we design different metrics for quantification. We use intrinsic information content (IIC) to measure an entity's uncertainty. In the taxonomy of entities, the closer an entity is to the root (i.e. the more abstract this entity is), the higher uncertainty it contains. We use domain and range to measure a relation's uncertainty: the richer entity types a relation links to, the more uncertainty it contains. Following [13], we represent each entity and relation by a Gaussian distribution, while the mean vector denotes its position and the diagonal covariance matrix denotes its uncertainty. To combine the element-level uncertainty into UKG embedding, we design a constraint between measurement and variance in Gaussian distribution and add it into parameter optimization process. We also use a metric-based framework, with the purpose of learning a similarity function that can effectively infer the true facts and their corresponding confidence scores given the few-shot support sets for each relation.

We conducted extensive experiments using two open uncertain knowledge graph datasets on two tasks: (i) tail entity prediction, which focuses on complete tail entities for the query; and (ii) confidence prediction, which seeks to predict confidence scores of unseen relation facts. Our method consistently outperforms the baseline models, justifying the efficacy of incorporation of the uncertainty information of entities and relations.

The rest of the paper is organized as follows. We first review the related work in Sect. 2, then provide the problem definition and propose our method in the next two sections. In Sect. 5, we present our experiments. Finally, we conclude the paper in Sect. 6.

2 Related Work

Here we survey three topics relevant to this work: deterministic KG embedding, uncertain KG embedding, and few-shot KG embedding.

2.1 Deterministic Knowledge Graph Embedding

Deterministic KG embedding methods have been extensively explored by recent works. There are two representative families of models, i.e. translation distance models and semantic matching models. For the former, a relation embedding is usually a transition or mapping for entity embeddings. Representative works include TransE [5], TransH [6], KG2E [14], and RotatE [15]. KG2E [14] represents entities and relations as multi-dimensional Gaussian distributions, but it cannot be used for embedding UKGs. For semantic matching methods, the scoring function evaluates the plausibility based on the latent semantics of entities given a triple. Representative works include RESCAL [16], DistMult [7], and ComplEx [8]. Recently, deep neural network based models like R-GCN [17] and KG-BERT [18] have been presented for further improvement.

2.2 Uncertain Knowledge Graph Embedding

UKGE [9] is the first work on embedding uncertain KGs, which utilizes a mapping function to transform plausibility scores to confidence scores and boosts its performance by applying probabilistic soft logic. GTransE [12] uses confidence-aware margin loss to deal with the uncertainty of triples in UKGs. PASSLEAF [10] extends UKGE for other types of scoring functions and includes pool-based semi-supervised learning to alleviate the false-negative problem for training.

2.3 Few-Shot Knowledge Graph Embedding

Recently, few-shot learning has been applied to KG completion. Xiong et al. [19] presented GMatching model for one-shot DKG embedding, which introduces a metric-based framework for relational learning. FSRL [20] proposed a more effective heterogeneous neighbor encoder under the few-shot learning setting. GMUC is the first work to study the few-shot UKG embedding problem, which utilizes metric-based learning to model entities and relations as multi-dimensional Gaussian distributions.

None of these models pays close attention to the uncertainty of entities and relations, which is the key problem we aim to solve in this paper.

3 Problem Definition

In this section, we formally define the uncertain knowledge graph embedding task and detail the corresponding few-shot learning settings.

3.1 Uncertain Knowledge Graph Embedding

An *uncertain knowledge graph* can be denoted by $\mathcal{G} = \{(h, r, t, s)|h, t \in \mathcal{E}, r \in \mathcal{R}, s \in [0, 1]\}$, where $\mathcal{E}$ is the entity set, $\mathcal{R}$ is the relation set, and s is the corresponding confidence score. Given an uncertain KG $\mathcal{G}$, the embedding model aims to encode each entity and relation in a low-dimensional space in which the structure information and confidence scores of facts are both preserved.

3.2 Few-Shot Learning Settings

In contrast to the most previous work [9,10,12] that usually assumes enough triples for each relation are available for training, this work studies the case where only few-shot triples (support set) are available. The goal of our work is to learn a metric that could be used to predict new facts with few examples.

Following the standard few-shot learning settings [21], we assume access to a set of training tasks. In our problem, each training task corresponds to a KG relation $r \in \mathcal{R}$, and has its own training/testing data: $T_r = \{S_r, Q_r\}$, where S_r is the support set for training, Q_r is the query set for testing. We denote this kind of task set as meta-training set, $\mathcal{T}_{meta-train}$. To imitate the few-shot relation prediction at evaluation period, there are only few-shot triples in each S_r. Besides, Q_r consists of the testing triples of r with ground-truth tail entities t_i and confidence scores s_i for each query (h_i, r), as well as the corresponding tail entity candidates $\mathcal{C}_{h_i,t} = \{t_{ij}\}$ where each t_{ij} is an entity in $\mathcal{G}$. The metric model can thus be tested on this set by ranking the candidates or predicting their confidence scores given the test query (h_i, r) and the support triples in S_r.

Once trained, we can use the model to predict on new relations, which is called the meta-testing step in literature. These meta-testing relations are unseen from meta-training. Each meta-testing relation also has its own few-shot training/testing data. These meta-testing relations form a meat-test set $\mathcal{T}_{meta-test}$.

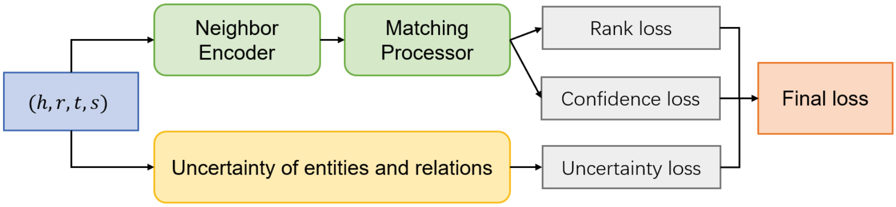

Fig. 1. The framework of our model.

Moreover, we leave out a subset of relations in $\mathcal{T}_{meta-train}$ as the meta-validation set $\mathcal{T}_{meta-val}$. Finally, we assume that the method has access to a background knowledge graph $\mathcal{G}'$, which is a subset of $\mathcal{G}$ with all the relations from $\mathcal{T}_{meta-train}$, $\mathcal{T}_{meta-test}$ and $\mathcal{T}_{meta-val}$ removed.

4 Methodology

In this section, we firstly introduce how we measure the uncertainty of entities and relations. We then present the framework for learning KG embedding and the learning strategy, as illustrated in Fig 1.

4.1 Uncertainty of Entities and Relations

We consider that the uncertainty of one entity/relation represents its semantic imprecision. The more imprecise semantic an entity/relation has, the higher uncertainty it contains. We utilize Intrinsic Information Content (IIC) to measure an entity's uncertainty, and utilize domain and range to measure a relation's uncertainty.

Uncertainty of Entities. IIC is used to measure an entity's uncertainty, and the lower IIC value one entity has, the more uncertainties it contains.

Information Content (IC) is an important dimension of word knowledge when assessing the similarity of two terms or word senses. The conventional way of measuring the IC of word senses is to combine knowledge of their hierarchical structure from an ontology like WordNet [22] with statistics on their actual usage in text as derived from a large corpus. However, IIC relies on hierarchical structure alone without the need for external resources. The calculation formula of IIC is defined as:

$$IIC(c) = 1 - \frac{\log(hypo(c) + 1)}{\log(N)} \tag{1}$$

where c is an arbitrary concept (essentially a node) in a taxonomy, N is a constant that is set to the maximum number of concepts that exist in the taxonomy, the function *hypo* returns the number of hyponyms of a given concept. The core idea behind IIC is that the more hyponyms a concept has the less information it expresses, otherwise there would be no need to further differentiate it.

Our method of obtaining an entity's uncertainty rests on the assumption that the taxonomic structure of entities in KG is organized in a meaningful and principled way, where entities near the root are more abstract than entities that are leaves. We argue that the more hyponyms an entity has, i.e. the lower IIC value, the higher uncertainty is contains. Considering the IIC value belongs to $[0, 1]$, we define the uncertainty value of an entity as:

$$UC_e(h) = 1 - IIC(h) \tag{2}$$

where h denotes an arbitrary entity in KG, $IIC(h)$ is its information content calculated by Eq. 1.

Uncertainty of Relations. We use domain and range to measure a relation's uncertainty. The richer the domain and range of a relation, the more uncertainties it contains.

RDF-Schema (RDFs) provides *rdfs:domain* and *rdfs:range* properties to declare the class of entities for relations. Domain restricts the class of head entities and range restricts the class of tail entities. We argue that if a relation links more entity classes, i.e. the domain and range are more diverse, it contains higher uncertainty. We introduce two different functions to calculate relation's uncertainties. One way utilizes the size of domain and range:

$$UC_r(r) = |D_r| \times |R_r| \tag{3}$$

where $|D_r|$ and $|R_r|$ are the size of domain and range set of relation r. The second one leverages the uncertainty of linked entity pairs:

$$UC_r(r) = \sum_{h \in D_r, t \in R_r} (UC_e(h) + UC_e(t)) \tag{4}$$

where h and t denote the entity class belongs to domain and range of relation r, respectively.

To incorporate the uncertainty of entities and relations into UKG embedding, we represent each of them as a multi-dimensional Gaussian distribution $\mathcal{N}(\mu, \Sigma)$, where $\mu \in \mathbb{R}^d$ is the mean vector indicating its position, $\Sigma = \sigma I$ $(\sigma \in \mathbb{R}^d)$ is the diagonal covariance matrix indicating its uncertainty. We denote it by $\mathcal{N}(\mu, \sigma)$ for convenience. We argue that the norm of variance vector is proportional to the uncertainty value for each entity/relation, and design a loss function $\mathcal{L}_{uc}$ as a constraint on variance vectors:

$$\mathcal{L}_{uc} = \sum_{i \in \mathcal{R}} \sum_{i \in \mathcal{E}} (w \cdot \|\sigma_i\|_2 + b - UC_{r/e}(i)) \tag{5}$$

where $\|\sigma_i\|_2$ is the l_2 norm of variance vector σ in Gaussian distribution, w and b are learnable parameters.

4.2 Model

Inspired by [20], we use a metric-based KG embedding framework composed of two major parts: (i) *Neighbor Encoder* utilizes local graph structure (i.e. one-hop neighbors) to enhance representation for each entity; (ii) *Matching Processor* calculates the similarity between a query and support set for relation prediction.

Neighbor Encoder. Considering different impacts of heterogeneous neighbors which may help improve entity embedding, we use heterogeneous neighbor encoder [20] to enhance the representation of given entity. Specifically, we denote the set of relational neighbors of given head entity h as $N_h = \{(r_i, t_i, s_i)|(h, r_i, t_i, s_i) \in \mathcal{G}'\}$, where $\mathcal{G}'$ is the background KG, r_i, t_i and s_i represent the i-th relation, corresponding tail entity and confidence score, respectively. The Neighbor Encoder should be able to encoder N_h and output a feature representation of h. We denote the output representation of h by $\mathcal{N}(\mathcal{F}^{\mu}_{NE}(h), \mathcal{F}^{\sigma}_{NE}(h))$, where $\mathcal{F}^{\mu}_{NE}(\cdot)$ and $\mathcal{F}^{\sigma}_{NE}(\cdot)$ are two heterogeneous neighbor encoders respectively for mean embedding and variance embedding.

By applying the neighbor encoder $\mathcal{F}_{NE}$ to each entity, we then concatenate head and tail entity embeddings to obtain the representation of each triple in the form of $\mathcal{N}(\mu_i, \sigma_i)$, where $\mu_i = [\mathcal{F}^{\mu}_{NE}(h) \oplus \mathcal{F}^{\mu}_{NE}(t)]$, $\sigma_i = [\mathcal{F}^{\sigma}_{NE}(h) \oplus \mathcal{F}^{\sigma}_{NE}(t)]$ and $\oplus$ is the concatenation operation. In this way, we can get the Gaussian representation of each query, $\mathcal{N}(\mu_q, \sigma_q)$. Since there are few-shot triples in support set, we utilize mean-pooling to aggregate them into one multi-Gaussian distribution $\mathcal{N}(\mu_s, \sigma_s)$.

Matching Processor. After the above operations, we can get two Gaussian distributions for each query and each support set. In order to measure the similarity between them, we employ the LSTM-based [23] recurrent processing block [24] to perform multi-step matching. We use two matching processors $\mathcal{F}^{\mu}_{MP}$ and $\mathcal{F}^{\sigma}_{MP}$ to calculate mean similarity sim_μ and variance similarity sim_σ respectively. To predict missing triples and their confidence scores, we define $s_{rank} = sim_\mu + \lambda sim_\sigma$ as ranking scores and $s_{conf} = sigmoid(w \cdot sim_\sigma + b)$ as confidence scores, where w and b are learnable parameters, and λ is a hyper-parameter.

4.3 Learning

For the query relation r, we randomly sample a set of few positive triples and regard them as the support set $\mathcal{S}_r = \{(h_i, t_i, s_i)|(h_i, r, t_i, s_i) \in \mathcal{G}\}$. The remaining positive triples are utilized as positive queries $\mathcal{Q}_r = \{(h_i, t_i, s_i)|(h_i, r, t_i, s_i) \in \mathcal{G} \cap (h_i, t_i, s_i) \notin \mathcal{S}_r\}$. Besides, we construct a group of negative triples $\mathcal{Q}'_r = \{(h_i, t'_i)|(h_i, r, t'_i, *) \notin \mathcal{G}\}$ by polluting the tail entities. Therefore, the ranking loss is formulated as:

$$\mathcal{L}_{rank} = \sum_r \sum_{(h,t,s)\in\mathcal{Q}_r} \sum_{(h,t',s')\in\mathcal{Q}'_r} s \cdot [\gamma + s_{rank} - s'_{rank}]_+ \tag{6}$$

where $[x]_+ = \max[0, x]$ is standard hinge loss and γ is margin distance, s_{rank} and s'_{rank} are rank scores between query $(h_i, t_i/t'_i)$ and support set $\mathcal{S}_r$.

To reduce the difference between the ground truth confidence score s_i and our predicting confidence score s_{conf}, we utilize mean squared error (MSE) between them as the MSE loss $\mathcal{L}_{mse}$. Specifically, $\mathcal{L}_{mse}$ is defined as:

$$\mathcal{L}_{mse} = \sum_r \sum_{(h_i,t_i,s_i)\in\mathcal{Q}_r} (s_{conf} - s_i)^2 \tag{7}$$

By leveraging the uncertainty loss $\mathcal{L}_{uc}$ of entities and relations, we define the final objective function as:

$$\mathcal{L}_{joint} = w_1\mathcal{L}_{rank} + w_2\mathcal{L}_{mse} + w_3\mathcal{L}_{uc} \tag{8}$$

where w_1, w_2 and w_3 are trade-off factors. To minimize $\mathcal{L}_{joint}$ and optimize model parameters, we take each relations as a task and design a batch sampling based meta-training procedure.

5 Experiments

In this section, we evaluate our models on two tasks: tail entity prediction and confidence score prediction. Ablation studies are followed to verify the impact of each module.

5.1 Datasets

The evaluation is conducted on two datasets named as NL27 k and CN15 k [9]. Table 2 gives the statistics of the datasets. NL27 k is extracted from NELL [3], an uncertain knowledge graph obtained from web pages. CN15 k is a subgraph of the common sense KG ConceptNet [4]. CN15 k matches the number of nodes with FB15 k - the widely used benchmark dataset for DKG embeddings, while NL27 k is a larger and more general dataset.

Entity's type and taxonomy structure are needed for calculating the uncertainty value of each entity/relation. For NL27 k, we utilize NELL's ontology to get the entity taxonomy. For CN15 k, since there is no ontology for ConceptNet, we align all entities in CN15k to DBpedia through the *ExternalURL* relation (one relation defined in ConceptNet) and string matching. Then, we leverage DBpedia's ontology to get the entity taxonomy. Meanwhile, both ontologies provide the type information for each entity. Then, we extract the domain and range of each relation from raw triple data.

We select the relations with less than 500 but more than 50 triples as few-shot tasks. There are 134 tasks in NL27 k and 11 tasks in CN15 k. In addition, we use 101/13/20 task relations for training/validation/testing in NL27k and the division is set to 8/1/2 in CN15 k. We refer the rest of the relations as background relations. According to Eq. 2, 3 and 4, we calculate the corresponding uncertainty values of entities and relations in both datasets and apply z-score normalization to maintain the scale consistency.

Table 2. Statistics of datasets. #Ent., #Rel., #Tri., #Task denote the number of entities, relations, triples, and tasks, respectively. Avg(s) and Std(s) are the average and standard deviation of the confidence scores, respectively.

Dataset	#Ent	#Rel	#Tri	#Task	Avg(s)	Std(s)
NL27k	27,221	404	175,412	134	0.797	0.242
CN15k	15,000	36	241,158	11	0.629	0.232

5.2 Baselines

In our comparison, we consider the following embedding-based methods: (i) UKGE, the first UKG embedding models, (ii) FSRL, a few-shot DKG embedding models, and (iii) GMUC, the first few-shot UKG embedding models. When evaluating UKGE, we use not only the triples of background relations but also all the triples of the training relations and the few-shot training triples of those in validate/test relations. When evaluating FSRL, we set a threshold $\tau = 0.75$ to distinguish high-confidence triples for training.

5.3 Experimental Setup

We tune hyper-parameters based weighted mean reciprocal rank (WMRR) on the validation datasets. The embedding dimension is set to 100 and 50 for NL27 k and CN15 k dataset, respectively. The maximum number of local neighbors in Neighbor Encoder is set to 30. In addition, the dimension of LSTM's hidden state is set to twice the embedding dimension. The number of matching steps equals 2. The margin distance γ is set to 5.0 for NL27 k and 6.0 for CN15 k. The initial learning rate equals 0.001 and the weight decay is 0.25 for each 10 k training steps. The batch size equals 256 and 64 for NL27 k and CN15 k. In entity candidate set construction, we set the maximum size to 1000 for both datasets. The few-shot size is set to 3 for the following experiments. The trade-off factors in the objective function are set to $w_1 = 1, w_2 = 1.1, w_3 = 0.01$.

5.4 Tail Entity Prediction

Tail entity prediction is a conventional evaluation task for knowledge graph embedding. The goal is to predict the tail entities given a head entity and a relation, which can be formulated as $(h, r, ?t)$.

Evaluation Protocol. Relations and their triples in training data are utilized to train the model while those of validation and test data are respectively used to tune and evaluate model. We use the top-k hit ratio (Hits@k), weighted mean rank (WMR), and mean reciprocal rank (WMRR) to evaluate performances of different methods. The k is set to 1, 5 and 10. The mean rank and mean reciprocal rank are linearly weighted by the confidence score.

Results. The performance of all models are reported in Table 3. We refer to the variant that adopt Eq. 3 to calculate relation's uncertainties as Ours_1 and name the one using Eq. 4 as Ours_2. We can see that our method produces consistent improvements over various embedding models on NL27 k and the superiority is lighter on CN15 k but still gets higher WMRR. UKGE performs much worse than the remaining models, proving that the metric-based framework is very effective in few-shot scenario. The Gaussian distribution representation is better than the point-based one, indicating that it is necessary to consider the uncertainty of entities and relations in UKGs. Our method outperforms GMUC, which demonstrates the rationality of the proposed uncertainty metrics for entities and relations. The results of Ours_1 are close to Ours_2, indicating that the two ways for calculating relation's uncertainties are both reasonable and effective. Comparing the model's performance on different datasets, all testing results on NL27 k are much better than those on CN15 k, which is caused by the metric-based learning process. Metric-based models aim to learn a matching function between support sets and queries to calculate their similarities, so they need (support set, query) pairs as many as possible, i.e., more training tasks. Nevertheless, CN15 k contains only 11 tasks, lesser than 134 tasks in NL27 k. Therefore, it not surprising that the evaluation results on CN15 k dataset are poor.

Table 3. Results of tail entity prediction

Dataset	Model	Hits@1	Hits@5	HIts@10	WMR	WMRR
NL27k	UKGE	0.031	0.038	0.046	489.537	0.037
	FSRL	0.216	0.373	0.490	81.728	0.294
	GMUC	0.363	0.549	0.626	65.146	0.455
	Ours_1	0.379	**0.598**	**0.670**	**50.940**	**0.481**
	Ours_2	**0.386**	0.573	0.663	51.539	0.474
CN15k	UKGE	**0.014**	0.019	0.028	496.185	0.022
	FSRL	0.006	0.025	0.041	374.439	0.023
	GMUC	0.002	0.027	0.089	382.188	0.027
	Ours_1	0.010	**0.042**	0.090	378.854	0.029
	Ours_2	0.013	0.037	**0.094**	**367.456**	**0.034**

5.5 Confidence Score Prediction

Confidence score prediction is to predict the confidence score given a triple, requiring the model to be uncertainty-aware. This task can be formulated as $(h, r, t, ?s)$.

Evaluation Protocol. For each uncertain relation fact (h, r, t, s) in the test query set, we predict the confidence score of (h, r, t) and report the mean squared error (MSE) and mean absolute error (MAE).

Results. Results are reported in Table 4. Since FSRL cannot predict the confidence scores of triples, we only compares with UKGE and GMUC. Our model also achieves the best performance in the this task. The results of UKGE are still worse, indicating that the metric-based few-shot KG embedding framework can also improve the accuracy of confidence prediction. Our method outperforms GMUC, illustrating that incorporating the uncertainty of entities and relations helps the model to predict the uncertainty of facts. The performances of Ours_1 and Ours_2 are very close, demonstrating the effectiveness of both methods for measuring the relation uncertainty. In the confidence prediction task, the results on two datasets are not much different, showing that confidence score information is easier to learn than the ranking information.

Table 4. Results of confidence score prediction

Dataset	NL27 k		CN15 k	
Metric	MSE	MAE	MSE	MAE
UKGE	0.468	0.636	0.350	0.541
GMUC	0.017	0.100	0.021	0.112
Ours_1	0.015	0.094	0.017	0.082
Ours_2	**0.015**	**0.092**	**0.017**	**0.079**

5.6 Ablation Studies

To investigate the contributions of different modules, we conduct the following ablation studies from three perspectives: (i) remove the Neighbor Encoder, denoted by No $\mathcal{F}_{NE}$, (ii) replace the LSTM-based Matching Network with the cosine similarity, denoted by No $\mathcal{F}_{MP}$, and (iii) exclude the uncertainty loss of entities and relations, denoted by No $\mathcal{L}_{uc}$.

Table 5. Results of ablation studies on different components.

Configuration	Hits@1	Hits@5	Hits@10	WMR	WMRR
No $\mathcal{F}_{NE}$	0.091	0.154	0.204	175.535	0.148
No $\mathcal{F}_{MP}$	0.129	0.239	0.317	118.207	0.203
No $\mathcal{L}_{uc}$	0.198	0.406	0.488	58.937	0.313
Ours_1	**0.234**	**0.450**	**0.540**	**49.093**	**0.346**

Table 5 shows the results of tail entity prediction in NL27k validate set and best results are highlighted in bold. We can see that removing any module will weaken the overall effect of our method, and the Neighbor Encoder has the greatest contribution. In addition, the results of No $\mathcal{L}_{uc}$ is worse than Ours_1, indicating that constraining the variance vector in Gaussian distribution by quantified element-level uncertainty can also boost the performance.

6 Conclusion

This paper introduces a few-shot uncertain knowledge graph embedding method by incorporating the uncertainty of entities and relations. We propose the corresponding metrics to measure the inherent element-level uncertainty of each entity and relation in UKGs and incorporate it into a few-shot metric-based framework to capture both semantic and uncertainty information. The extensive experiments on two public datasets demonstrate that our proposed method can outperform the state-of-the-art baseline models. Our future work might consider extending the metrics for element-level uncertainty into deterministic knowledge graphs to improve the learned embeddings.

Acknowledgements. This work is supported by the NSFC (Grant No. 62006040), the Project for the Doctor of Entrepreneurship and Innovation in Jiangsu Province (Grant No. JSSCBS20210126), the Fundamental Research Funds for the Central Universities, and ZhiShan Young Scholar Program of Southeast University.

References

1. Mao, Z., Wang, B., Guo, L., Wang, Q.: Knowledge graph embedding: a survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**(12), 2724–2743 (2017)
2. Wu, W., Li, H., Wang, H., Zhu, K.: Probase: for text understanding. In: Proceedings of SIGMOD, pp. 481–492 (2012)
3. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: Proceedings of AAAI (2010)
4. Speer, R., Chin, J., Havasi, C.: ConceptNet 5.5: an open multilingual graph of general knowledge. In: Proceedings of AAAI, pp. 4444–4451 (2017)
5. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of NeurIPS, pp. 2787–2795 (2013)
6. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of AAAI, pp. 1112–1119 (2014)
7. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of ICLR (Poster) (2015)
8. Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of ICML, pp. 2071–2080 (2016)
9. Chen, X., Chen, M., Shi, W., Sun, Y., Zaniolo, C.: Embedding uncertain knowledge graphs. In: Proceedings of AAAI, pp. 3363–3370 (2019)

10. Chen, Z., Yeh, M., Kuo, T.: PASSLEAF: a pool-based semi-supervised learning framework for uncertain knowledge graph embedding. In: Proceedings of AAAI, pp. 4019–4026 (2021)
11. Chen, X., Boratko, M., Chen, M., Dasgupta, S., Li, X., McCallum, A.: Probabilistic box embeddings for uncertain knowledge graph reasoning. In: Proceedings of NAACL-HLT, pp. 882–893 (2021)
12. Kertkeidkachorn, N., Liu, X., Ichise, R.: GTransE: generalizing translation-based model on uncertain knowledge graph embedding. In: Proceedings of JSAI, pp. 170–178 (2019)
13. Zhang, J., Wu, T., Qi, G.: Gaussian metric learning for few-shot uncertain knowledge graph completion. In: Jensen, C.S., et al. (eds.) DASFAA 2021. LNCS, vol. 12681, pp. 256–271. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-73194-6_18
14. He, S., Liu, K., Ji, G., Zhao, J.: Learning to represent knowledge graphs with Gaussian embedding. In: Proceedings of CIKM, pp. 623–632 (2015)
15. Sun, Z., Deng, Z., Nie, J., Tang, J.: Rotate: knowledge graph embedding by relational rotation in complex space. In: Proceedings of ICLR (Poster) (2019)
16. Nickel, M., Tresp, V., Kriegel, H.: A three-way model for collective learning on multi-relational data. In: Proceedings of ICML, pp. 809–816 (2011)
17. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
18. Yao, L., Mao, C., Luo, Y.: KG-BERT: BERT for knowledge graph completion. CoRR abs/1909.03193 (2019)
19. Xiong, W., Yu, M., Chang, S., William Yang Wang, X.G.: One-shot relational learning for knowledge graphs. In: Proceedings of EMNLP, pp. 1980–1990 (2018)
20. Zang, C., Yao, H., Huang, C., Jiang, M., Li, Z., Chawla, N.V.: Few-shot knowledge graph completion. In: Proceedings of AAAI, pp. 3041–3048 (2020)
21. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: Proceedings of ICLR (2017)
22. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
23. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
24. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching networks for one shot learning. In: Proceedings of NeurIPS, pp. 3630–3638 (2016)