Relation-Aware Network with Attention-Based Loss for Few-Shot Knowledge Graph Completion

Qiao Qiao✉, Yuepei Li, Kang Zhou, and Qi Li[0000−0002−3136−2157]

Iowa State University, Ames, Iowa, USA
{qqiao1,liyp0095,kangzhou,qli}@iastate.edu

Abstract. Few-shot knowledge graph completion (FKGC) task aims to predict unseen facts of a relation with few-shot reference entity pairs. Current approaches randomly select one negative sample for each reference entity pair to minimize a margin-based ranking loss, which easily leads to a zero-loss problem if the negative sample is far away from the positive sample and then out of the margin. Moreover, the entity should have a different representation under a different context. To tackle these issues, we propose a novel Relation-Aware Network with Attention-Based Loss (RANA) framework. Specifically, to better utilize the plentiful negative samples and alleviate the zero-loss issue, we strategically select relevant negative samples and design an attention-based loss function to further differentiate the importance of each negative sample. The intuition is that negative samples more similar to positive samples will contribute more to the model. Further, we design a dynamic relation-aware entity encoder for learning a context-dependent entity representation. Experiments demonstrate that RANA outperforms the state-of-the-art models on two benchmark datasets.

Keywords: Few-shot learning · Knowledge graph completion

1 Introduction

Knowledge graphs (KGs) contain rich triples (facts), where each triple (h, r, t) illustrates a relation r between a head entity h and a tail entity t. KGs such as Wikidata [16] and NELL [3] have been applied to various downstream applications such as relation extraction [25], named entity recognition [24], and node classification [10].

Knowledge Graph Completion (KGC) is proposed to solve the issue of incompleteness caused by missing entities or relations in the KGs. KG embedding [2, 15] has achieved considerable performance on KGC. These models perform well with enough training triples, but a large portion of relations in KGs follow a long-tail distribution. For example, around 10% of relations in Wikidata [4] have no more than 10 triples. Relations that do not have enough training triples are known as few-shot relations. It is crucial and challenging for the model to predict relations with limited training triples.

Few-shot knowledge graph completion (FKGC) methods have been proposed to address the few-shot relation issue. Given the relation r and few-shot reference entity pairs (h, t), the FKGC aims to rank candidate tail entities t for each query $(h, ?)$. These few-shot reference entity pairs form a support set, and queries form a query set. One line of the existing methods focuses on designing metric learning algorithms to compute the

similarity between entity pairs [18, 23]. Another line leverages model agnostic meta-learning algorithm (MAML) [5] to learn the optimal parameters of the model [4, 9, 17].

To train the model, current FKGC methods apply a margin-based ranking loss function that aims to separate the score of the positive triple from the score of the negative triple by a margin. One negative triple is formed for each positive triple by replacing the true tail entity with a randomly selected candidate tail entity. This loss function does not effectively utilize the negative samples. Furthermore, an irrelevant negative sample is likely to be selected due to a large number of candidates. These irrelevant negative samples lead to zero loss because the negative triple is far away from the positive triple. Therefore these irrelevant negative samples would not contribute to the training and slow down the convergence rate [11]. For example, given a true triple *(Kobe Bryant, WorkIn, California)*, the model can select negative tail entities, such as *New York, Thailand, London*, etc. Because *Thailand* is irrelevant to the true tail entity *California*, the distance between *California* and *Thailand* is greater than a predefined margin, and the corresponding loss is zero. Thus *Thailand* may not contribute to the training.

To address the above limitations, we propose a framework called RANA (**R**elation-**A**ware **N**etwork with **A**ttention-Based Loss). To improve the quality of negative samples, we propose to filter irrelevant candidate tail entities first and then randomly sample multiple negative samples instead of one. Since the importance of negative samples is different and depends on their similarities to the positive sample, we apply an attention mechanism to assign a weight to each negative sample, where the weights of the most relevant negative samples are higher than the weights of the less relevant negative samples. The attention-based weighted loss function can enable the model to effectively avoid zero-loss issues and thus learn a better decision boundary.

Further, we propose a context-dependent dynamic relation-aware entity encoder to learn different representations of an entity in different relations. Specifically, given a target relation and its support set, the entity encoder uses the similarities between the target relation and neighboring relations to differentiate the impact of neighboring entities and dynamically encode the local connections of the entity. Finally, RANA employs meta-learning to enable the model to perform well on a new relation with a few training triples in a small number of gradient steps.

In summary, our main contributions are:

1. We propose a new negative sampling strategy and a novel attention-based loss function to solve the zero-loss and slower convergence issues.
2. We propose a dynamic entity encoder to learn a context-dependent entity representation and reduce the influence of unrelated neighboring entities.
3. Experiment results on benchmark datasets show that RANA consistently and significantly outperforms other baseline methods.

2 Related Work

2.1 Embedding based Knowledge Graph Completion

Knowledge graph embedding aims to embed entities and relations into a low-dimensional continuous vector space while preserve their semantic meaning. Existing methods can

be divided into the following categories: (1) Translation-based models calculate the Euclidean distance between entities and relations as the plausibility of a fact, such as TransE [2], RotatE [13], and TransMS [20]; (2) Semantic matching-based models calculate the semantic similarity between entities and relations as the plausibility of a fact, such as RESCAL [8], DistMult [19], and PUDA [14]; and (3) Neural network-based models take entities and relations into a deep neural network to fuse the graph network structure and content information of entities and relations, such as SME [1], CompLEx [15], and BertRL [22]. All above models require sufficient training triples and thus impair their performance on few-shot relations.

2.2 Few-Shot Knowledge Graph Completion

FKGC requires the model to predict new facts with a few training facts. Existing methods fall into two categories: (1) Metric-based models aim to learn the matching metrics by calculating the similarity between the query set and the support set. GMathching [18] focuses on one-shot KGC by considering both the learned embeddings and local graph structures. FSRL [23] and FAAN [12] extend GMatching to few-shot scenarios. (2) Optimization-based models aim to learn a set of good initial model parameters so that the learned model can be generalized to the new relation quickly. MetaR [4], GANA [9], and HiRe [17] focus on extracting relation-specific meta information from the embeddings of entities in the support set and transferring it to the query set.

However, all these methods use a margin-based ranking loss, which can not effectively avoid the low-quality negative sample, leading to a zero-loss issue and influencing the convergence rate. Negative sampling has been proven as important as positive sampling in determining the optimization objective [21]. Especially under the few-shot setting, given limited positive samples, how to select high-quality negative samples based on the corresponding positive sample is crucial.

3 Preliminary

3.1 Problem Definition

Knowledge Graph $\mathcal{G}$. A knowledge graph $\mathcal{G}$ is a set of triples $\mathcal{T} = \{(h, r, t) \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}\}$, where $\mathcal{E}$ and $\mathcal{R}$ represent the entity set and relation set, respectively. The relation set $\mathcal{R}$ contains few-shot relations and high-frequency relations. The background knowledge graph $\mathcal{G}_{background}$ is a set of triples associated with all high-frequency relations.
Knowledge Graph Completion. The KGC task is to either predict the tail entity t given the head entity h and the query relation r: $(h, r, ?)$ or predict unseen relation r between two existing entities: $(h, ?, t)$. In this work, we focus on tail entity prediction.
Few-shot Knowledge Graph Completion. Given a relation $r \in \mathcal{R}$ and its few-shot support set $\mathcal{S} = \{(h_i, t_i) \in \mathcal{T}\}$, the FKGC task aims to predict tail entity t for each query $\mathcal{Q} = \{(h_i, ?) \in \mathcal{T}\}$.
A Few-shot Relation's Neighborhood. Given a triple (h, r, t) of a few-shot relation r, the neighborhood of r is defined as $\{h, t, \mathcal{N}_h, \mathcal{N}_t\}$, where $\mathcal{N}_h$ and $\mathcal{N}_t$ are the sets of one-hop neighbors of h, t, respectively. All $\mathcal{N}_h$ and $\mathcal{N}_t$ are from the background

knowledge graph $\mathcal{G}_{background}$. A neighbor in $\mathcal{N}_h$ or $\mathcal{N}_t$ is composed of a neighboring relation r_i and a neighboring entity c_i. We denote the neighbor of each entity (h or t) as $\mathcal{N}_e = \{(r_i, c_i)|(e, r_i, c_i) \in \mathcal{G}_{background}\}$.

3.2 Meta-learning Settings

Meta-learning aims to train a model on several related tasks so that the model can quickly learn a new task using a few training data. We leverage an optimization-based meta-learning algorithm called MAML [5], which aims to learn a task-specific parameter set $\mathbf{\Theta}_i$ by using well-initialized meta-model parameter set $\mathbf{\Theta}$. It can be divided into two stages, meta-training and meta-testing. During meta-training, given a task $\mathcal{T}_i$, a support set $\mathcal{S}_i$ and a query set $\mathcal{Q}_i$ are first sampled from $\mathcal{T}_i$. Then, the model learns a task-specific parameter set $\mathbf{\Theta}_i$ by one gradient descent update on the support set $\mathcal{S}_i$:

$$\mathbf{\Theta}_i = \mathbf{\Theta} - \eta * \nabla \mathcal{L}_{\mathcal{S}_\mathbf{i}}(\mathbf{\Theta}). \tag{1}$$

Finally, meta-optimization across all query sets of tasks is performed to learn the meta-model parameter set $\mathbf{\Theta}$ by using task-specific parameter set $\mathbf{\Theta}_i$. During meta-testing, the model can quickly adapt to a new task using only a support set $\mathcal{S}$.

In FKGC, each task is defined as predicting new triples for a specific few-shot relation. All the relations in the meta-training form a meta-training set $\mathcal{R}_{meta-training}$. Since the goal is to predict facts of unseen relations, the relations in meta-validation $\mathcal{R}_{meta-validation}$, meta-testing $\mathcal{R}_{meta-testing}$, and $\mathcal{R}_{meta-training}$ are distinct.

4 Methodology

In this section, we first introduce triple representation, which aims to learn a context-dependent entity representation and a good initialization few-shot relation representation. Then we introduce a novel negative sampling strategy, which aims to filter irrelevant candidate tail entities and use an attention mechanism to differentiate the importance of each negative sample. Finally, we introduce meta-learning, which aims to learn well-generalized parameters so that the model can quickly adapt to a new task using few reference triples. Fig.1 shows the framework of RANA for a few-shot relation *WorkIn*.

4.1 Triple Representation

Dynamic Relation-Aware Entity Encoder. The entity representation should be context-dependent. For example, *(Kobe Bryant, California)* can involve in two different relations, such as *WorkIn* and *DieIn*, so *Kobe Bryant* should have different embeddings in these two different relation contexts.

Besides, given a few-shot relation, different neighbors should have a different impact on the entity itself. For example, in Fig.1, given the few-shot relation *WorkIn* and the head entity *Kobe Bryant*, its neighbor *(AthleteOf, Lakers)* should get more attention since it reveals work information about *Kobe Bryant*, but the neighbor *(HasSpouse, Vanessa)* should get less attention since it reveals family information of *Kobe Bryant* which is irrelevant to the few-shot relation *WorkIn*.

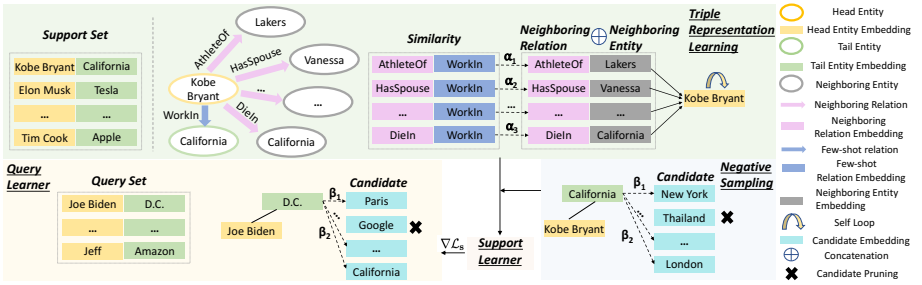


Fig. 1: The framework of RANA for a few-shot relation *WorkIn*

To address these issues, we design a dynamic relation-aware entity encoder, which incorporates neighboring relations to learn different embeddings of an entity in different relations and differentiates the importance of each neighbor by an attention mechanism.

Given an entity pair (h, t) from a support set $\mathcal{S}$, the embedding of few-shot relation r is defined as:

$$\mathbf{r} = \mathbf{t} - \mathbf{h}, \tag{2}$$

where $\mathbf{h}$ and $\mathbf{t}$ are the pretrained embeddings by TransE [2].

Here, we use the head entity h as an example to illustrate the entity encoding procedure, and this procedure also holds for the tail entity t.

To differentiate the impact of each neighbor, we use a Multilayer Perceptron (MLP) network to calculate the relevance score between the few-shot relation r and each neighboring relation r_i.

The relevance score is defined as follows:

$$m(r, r_i) = \mathbf{W_2}[\mathbf{tanh}(\mathbf{W_1}[\mathbf{r} \oplus \mathbf{r_i}])], \tag{3}$$

where $\oplus$ denotes the concatenation operation, $\mathbf{r_i}$ denotes the embedding of neighboring relations, and $\mathbf{W_1}$ and $\mathbf{W_2}$ are trainable parameters. A higher relevance score between the neighboring relation and the few-shot relation means that this neighboring relation is more important to the few-shot relation.

To learn the different representations of an entity in different relations, we design a dynamic neighbor embedding $\mathbf{A_{r_i,c_i}}$ of the head entity h as follows:

$$\mathbf{A_{r_i,c_i}} = \sum_{(r_i,c_i)\in\mathcal{N}_h} \alpha_i \mathbf{W_3}[\mathbf{r_i} \oplus \mathbf{c_i}], \tag{4}$$

where $\mathbf{W_3}$ are trainable parameters, and α_i is the attention score of each neighbor:

$$\alpha_i = \frac{exp(m(r, r_i))}{\sum_{r_i\in\mathcal{N}_h} exp(m(r, r_i))}. \tag{5}$$

When the neighboring relation is more relevant to the few-shot relation, the higher attention α_i is given to the corresponding neighbor. Then this neighbor will play a more important role in neighbor embedding.

Since the information of entity h itself is still valuable, we combine the embedding of entity h with $\mathbf{A}_{\mathbf{r_i},\mathbf{c_i}}$ to get the final representation $\mathbf{h}'$ as follows:

$$\mathbf{h}' = \sigma(\mathbf{W_4}(\mathbf{h} + \mathbf{A}_{\mathbf{r_i},\mathbf{c_i}})), \tag{6}$$

where $\mathbf{W_4}$ are trainable parameters and $\sigma(\cdot)$ is a sigmoid function.

Few-Shot Relation Representation. The same entity pair may involve in different relations, so the learning of relation representation is necessary, and it can further help triple representation learning.

The relation representation from a specific entity pair in the support set $\mathcal{S}$ is:

$$\mathbf{R}_{(\mathbf{h_i},\mathbf{t_i})} = FC^{\sigma}_{\mathbf{W_5}}[\mathbf{h'_i} \oplus \mathbf{t'_i}], \tag{7}$$

where the fully connected layer $FC^{\sigma}_{\mathbf{W_5}}$ is parameterized by $\mathbf{W_5}$ and activated by a LeakyReLU function $\sigma(\cdot)$.

The relation representation from the support set $\mathbf{R^s}$ is then the average of all representations from entity pairs in $\mathcal{S}$,

$$\mathbf{R^s} = \frac{\sum_{i=1}^{I} \mathbf{R}_{(\mathbf{h_i},\mathbf{t_i})}}{I}, \tag{8}$$

where I is the number of entity pairs in the support set $\mathcal{S}$.

4.2 Negative Sampling

Since the positive sample is limited under the few-shot setting, how to take advantage of negative samples is more critical. Previous FKGC methods use a margin-based ranking loss and randomly select one negative sample for each positive sample [18, 4, 23, 12, 9]. But the random selection is likely to select an irrelevant negative sample and lead to a zero-loss issue. Further, regardless of their relevance to the positive samples, all negative samples will have the same impact on the model training. To address these issues, RANA filters irrelevant negative samples and uses an attention mechanism to distinguish the importance of each negative sample.

Candidate Pruning. The candidate set of negative samples constructed by [18] limits the candidate entities to those have the same types as the true tail entities in the support set, but this broad candidate set includes many irrelevant candidates as negative samples. For example, given a fact *(Kobe Bryant, WorkIn, California)*, the previous candidate set is limited to location and company types of entities because the types of tail entities in the support set are company or location. However, a candidate such as *Thailand* is irrelevant to *California* and thus is not helpful in the model training.

To reduce the number of irrelevant candidates and enable the model to select high-quality negative samples during the training stage, RANA filters irrelevant candidates by the similarity of the true tail entity t and a candidate tail entity t^-. The similarity is calculated by:

$$f(\mathbf{t}, \mathbf{t}^-) = \mathbf{t}^{-\mathbf{T}}\mathbf{t}, \tag{9}$$

where $\mathbf{t}$ is the embedding of a true tail entity and $\mathbf{t}^-$ is the embedding of a candidate tail entity. If $f(\mathbf{t}, \mathbf{t}^-) < \tau$, where τ is a threshold, then t^- should be filtered.

Attention of Negative Samples. To fully utilize the negative samples, RANA selects multiple negative samples instead of one and differentiates each negative triple's contribution by an attention mechanism.

Intuitively, if a negative sample is more relevant to the positive sample, this negative sample should play a more important role in model training. Therefore, higher attention should be given to this negative sample. As shown in Fig.1, given a positive sample *(Kobe Bryant, California)*, the negative sample *(Kobe Bryant, New York)* is more relevant to the positive sample than the negative sample *(Kobe Bryant, London)*, and thus the model should pay more attention to the former.

We define a scaled-dot product function $f(\mathbf{p_i}, \mathbf{n_{ij}})$ to calculate the similarity between the positive sample (h_i, t_i) and each of its negative sample (h_i, t_{ij}^-):

$$\mathbf{p_i} = \mathbf{h_i} \oplus \mathbf{t_i}, \qquad \mathbf{n_{ij}} = \mathbf{h_i} \oplus \mathbf{t_{ij}^-}, \qquad f(\mathbf{p_i}, \mathbf{n_{ij}}) = \frac{\mathbf{n_{ij}^T p_i}}{\sqrt{|p|}}, \tag{10}$$

where $|p|$ is the dimension of $\mathbf{p_i}$. The attention of each negative triple is defined by:

$$\beta_{ij} = \frac{exp\, f(\mathbf{p_i}, \mathbf{n_{ij}})}{\sum_{j=1}^{J} exp\, f(\mathbf{p_i}, \mathbf{n_{ij}})}, \tag{11}$$

where J is the number of negative samples.

The Loss of RANA. Negative sampling is as valuable as positive sampling in determining the optimization object, but it has been overlooked in the margin-based ranking loss [21]. To alleviate zero-loss and slower convergence issue, we sample multiple negative triples instead of one to increase the probability of generating a relevant negative triple.

Motivated by TransE [2], we first calculate the distance of each entity pair (h_i, t_i) as follows:

$$d_{(h_i,t_i)} = ||\mathbf{h_i} + \mathbf{R} - \mathbf{t_i}||_{L2}, \tag{12}$$

Because the smaller distance indicates the triple is more likely to be true, the triple should lead to a higher score. The score function of each triple is designed as:

$$s_{(h_i,t_i)} = \gamma - d_{(h_i,t_i)}, \tag{13}$$

where γ is a hyperparameter.

Our log-based loss function is:

$$\mathcal{L} = -\sum_{i=1}^{I} log\, \sigma(s_{(h_i,t_i)}) - \sum_{i=1}^{I}\sum_{j=1}^{J} \beta_{ij} log\, \sigma(-s_{(h_i,t_{ij}^-)}), \tag{14}$$

where $\sigma(\cdot)$ is a sigmoid function, and β_{ij} is the attention of each negative triple calculated by Eq.(11). Since a more relevant negative triple has higher attention (β_{ij}), this loss function will make those relevant negative triples impact more in model training.

4.3 Meta Learning

To learn a new relation quickly with a support set, RANA employs MAML [5] to optimize the model parameters that can be adapted for few-shot relations.

Algorithm 1 Training framework

Input: Training tasks $\mathcal{R}_{meta-training}$, initial model parameter $\mathbf{\Theta}$
Pre-trained KG embedding (excluding relation in $\mathcal{R}_{meta-training}$)

while not done **do**
  Sample a task $\mathcal{T}_i = \{\mathcal{S}_i, \mathcal{Q}_i\}$ from $\mathcal{R}_{meta-training}$
  Get $\mathbf{R^s}$ from $\mathcal{S}_i$ by Eq.(2)-Eq.(8)
  Get negative sample of $\mathcal{S}_i$ by Eq.(9)-Eq.(11)
  Calculate the loss of $\mathcal{S}_i$ by Eq.(12)-Eq.(14)
  Update the embedding of the task-specific relation $\mathbf{R^q}$ with gradient descent by Eq.(15)
  Get negative samples of $\mathcal{Q}_i$ by Eq.(9)-Eq.(11)
  Calculate the loss of $\mathcal{Q}_i$ by Eq.(12)-Eq.(14)
  Update whole model parameters $\mathbf{\Theta} \leftarrow \mathbf{\Theta} - \mu \nabla \mathcal{L}$
end while

Support Learner. Support learner aims to learn a representation $\mathbf{R^s}$ of the few-shot relation and $\mathbf{R^s}$ can be calculated by Eq.(2)-Eq.(8).

Query Learner. Following the MetaR [4] assumption, the relation is the key common information between support and query set. So we aim to transfer the support relation R^s to the query relation R^q by minimizing a loss function via gradient descending.

In RANA, the relation embedding $\mathbf{R^q}$ can be updated by the gradient descent,

$$\mathbf{R^q} = \mathbf{R^s} - \eta * \nabla \mathcal{L}_\mathbf{s}, \tag{15}$$

where the hyperparameter η refers to the step size and $\mathcal{L}_s$ refers to the loss of the corresponding support set, which is calculated by Eq.(12)-Eq.(14).

To update all parameters of RANA, we use the updated relation embedding $\mathbf{R^q}$ to calculate the loss of the corresponding query set $\mathcal{L}_q$ by Eq.(12)-Eq.(14) as well.

Objective and Training Process. During the meta training-stage, the objective of RANA is to minimize the sum of query loss for all tasks, and the overall loss is:

$$\mathcal{L} = arg \min_{\mathbf{\Theta}} \sum \mathcal{L}_q, \tag{16}$$

where $\mathbf{\Theta}$ represents all trainable parameters.

4.4 Algorithm of RANA

We summarize the overall training procedure in Algorithm 1.

4.5 Difference from RotatE

RotatE [13] is an embedding-based KGC method that uses a self-adversarial negative sampling technique to effectively optimize the model. Our approach differs from RotatE in a major way: We consider the similarity between the positive triple and negative triple as the weight of each negative triple, but RotatE considers the distribution of negative triples and treats the probability as the weight of each negative triple. Therefore, the weights of the negative samples in RotatE are independent of the positive samples. As we will show in the experiments (section 5.5), RANA can achieve a better performance than RotatE's self-adversarial negative sampling under the few-shot setting.

Table 1: Statistics of the Datasets. Columns 2-7 represent the number of entities, relations, triples, relations in $\mathcal{R}_{meta-training}$, relations in $\mathcal{R}_{meta-validation}$, and relations in $\mathcal{R}_{meta-testing}$, respectively.

Dataset	#Ent	#Rel	#Triples	#Train Rel	#Valid Rel	#Test Rel
NELL-One	$68,545$	358	$181,109$	51	5	11
Wiki-One	$4,838,244$	822	$5,859,240$	133	16	34

5 Experiments

5.1 Datasets and Evaluation Metrics

We conduct experiments on NELL-One and Wiki-One, constructed by [18]. In both datasets, relations with more than 50 but less than 500 triples are selected as few-shot relations, and the remaining relations are treated as background relations. We use 51/5/11 and 133/16/34 few-shot relations for training/validation/testing in NELL-One and Wiki-One, respectively. The statistics of both datasets are shown in Table 1.

To evaluate the performance of RANA and all baselines, we utilize two metrics: mean reciprocal rank (MRR) and Hits@K. MRR is the mean reciprocal rank of correct entities, and Hits@K is the proportion of correct entities ranked in the top k.

5.2 Baseline

Traditional embedding-based methods aim to learn entity and relation embeddings by modeling relational structure in KG. We consider the following widely used methods as baselines: TransE [2], DistMult [19], ComplEx [15], SimplE [6], and RotatE [13]. All these methods require sufficient training triples for each relation and do not use local graph structure to update entity embeddings.

FKGC methods aim to learn long-tail and unseen relations by utilizing deep neural networks to explore the connection between the support set and the query set. We consider the following models as baselines: GMatching [18], MetaR [4], FSRL [23], FAAN [12], GANA [9], and HiRe [17]. We run RANA 5 times and report the average results.

5.3 RANA Setups

The pre-trained entity and relation embeddings are obtained from TransE. The embedding dimension is set to 50 and 100 for NELL-One and Wiki-One, respectively. We use Adam [7] with the initial learning rate of 0.01 to update parameters. The number of negative samples is 5, the margin γ is 12.0, the step size η is 1, and the number of neighbors is 25 on both datasets. The model with the highest MRR on the validation set is applied as the final model. The optimal hyperparameters are tuned on the validation set by grid search. We conduct RANA on a server with a Tesla V100 GPU (32G).

5.4 Overall Evaluation Results and Analysis

The performances of all models on NELL-One and Wiki-One are reported in Table 2. Compared to the traditional embedding-based methods, incorporating graph neighbors

Table 2: Results of 5-shot KGC. **Bold** numbers represent the best results and underline numbers denote the runner-up results. † cites the result from [12], $*$ cites the result from their original papers.

	NELL-One				Wiki-One			
Model	MRR	Hits@10	Hits@5	Hits@1	MRR	Hits@10	Hits@5	Hits@1
TransE†	0.174	0.313	0.231	0.101	0.133	0.187	0.157	0.100
DistMult†	0.200	0.311	0.251	0.137	0.071	0.151	0.099	0.024
ComplEx†	0.184	0.297	0.229	0.118	0.080	0.181	0.122	0.032
SimplE†	0.158	0.285	0.226	0.097	0.093	0.180	0.128	0.043
RotatE†	0.176	0.329	0.247	0.101	0.049	0.090	0.064	0.026
GMatching†	0.176	0.294	0.233	0.113	0.263	0.387	0.337	0.197
MetaR†	0.209	0.355	0.280	0.141	0.323	0.418	0.385	0.270
FSRL†	0.153	0.319	0.212	0.073	0.158	0.287	0.206	0.097
FAAN†	0.279	0.428	0.364	0.200	0.341	0.463	0.395	0.281
GANA*	0.344	0.517	0.437	0.246	0.351	0.446	0.407	0.299
HiRe*	0.306	0.520	0.439	0.207	0.371	0.469	0.419	0.319
RANA	**0.361±0.011**	**0.573±0.009**	**0.475±0.010**	**0.253±0.013**	**0.379±0.008**	**0.480±0.012**	**0.437±0.008**	**0.329±0.011**

is effective for learning entity embedding. RANA outperforms the other FKGC models on both datasets. Compared with the runner-up results, the improvements obtained by RANA in terms of MRR, Hits@10, Hits@5, and Hits@1 are 4.9%, 10.2%, 8.2%, 2.8% on NELL-One, and 2.2%, 2.3%, 4.3%, 3.1% on Wiki-One, respectively.

5.5 Ablation Study

RANA is composed of two modules, including a dynamic relation-aware entity encoder and negative sampling. To investigate the contributions of each component, we conduct the 5-shot KGC with different settings. The results are summarized in Table 3.

Entity Encoder Variants: We analyze the impact of the neighboring relation in Eqs.(4) and (5) by removing r_i from Eq.(4) or adding c_i in Eq.(5). Besides, we remove the attention mechanism in Eq.(4). The results show that neighboring relation and attention mechanism can benefit model performance. It illustrates semantic information of relations can improve the entity representation, and different relations should have different impacts on the entity itself. Since the effect of the attention mechanism depends on neighbors, Wiki-One has much sparser neighbors than NELL-One [9], so the attention mechanism plays a small role in Wiki-One.

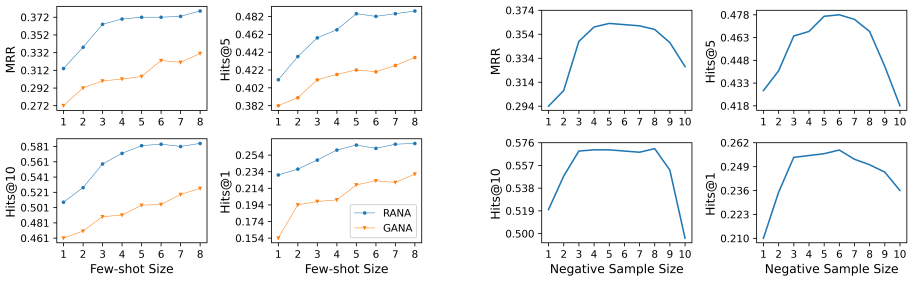
Negative Sampling Variants: To inspect the effectiveness of the negative sampling and attention-based loss functions, we conduct five different experiments. (A) We use only one negative sample in Eq.(14). (B) We remove the negative attention mechanism in Eq.(14). (C) We remove the candidate pruning stage. (D) We remove the candidate pruning stage and negative attention mechanism. (E) We replace Eq.(14) with RotatE [13] self-adversarial negative sampling loss. Experimental results show that the negative sampling strategy plays a key role in the success of RANA.

5.6 Influence of Size of Few-shot Support Set and Negative Sample

To analyze the impact of support set size, we compare RANA with GANA on NELL-one. Fig.2a shows the performances with support set size from 1 to 8. RANA outperforms GANA under different sizes of support sets, showing the effectiveness of RANA.

Table 3: Ablation Study

Model	NELL-One				Wiki-One			
	MRR	Hits@10	Hits@5	Hits@1	MRR	Hits@10	Hits@5	Hits@1
whole model	**0.372**	**0.580**	**0.477**	**0.257**	**0.387**	**0.486**	**0.443**	**0.339**
Eq.(4) w/o r_i	0.339	0.535	0.427	0.222	0.362	0.468	0.410	0.299
Eq.(5) with c_i	0.358	0.573	0.471	0.256	0.367	0.477	0.424	0.302
Eq.(4) w/o α_i	0.326	0.526	0.407	0.235	0.377	0.483	0.433	0.315
Eq.(14) with one negative sample	0.294	0.520	0.428	0.210	0.349	0.451	0.417	0.311
Eq.(14) w/o negative attention	0.293	0.494	0.416	0.213	0.298	0.387	0.371	0.257
w/o candidate pruning	0.298	0.507	0.425	0.217	0.311	0.445	0.360	0.243
w/o candidate pruning and negative attention	0.257	0.447	0.396	0.192	0.286	0.363	0.321	0.242
RotatE self-adversarial negative sampling	0.268	0.479	0.365	0.165	0.310	0.389	0.401	0.255



(a) (b)

Fig. 2: (a) Influence of Few-shot Support Set Size,(b) Influence of Negative Sample Size

After the 5-shot, the improvement of RANA is not significant. We randomly select 20 facts from the relation *teamcoach* to analyze the errors in the 5-shot setting. RANA predicts 12 out of 20 true tail entities in top 10. Among the other 8 facts, 4 of them have incorrect ground truth tail entities, and 3 of them have neighbors fewer than 10. For these cases, increasing the size of the support set is unlikely to change the results.

We conduct an experiment to analyze the influence of the negative sample size. Fig.2b shows the performance of RANA on NELL-One with the negative sample size from 1 to 10. The performance improves initially when increasing the negative sample size. After size 6, the performance begins to drop due to the class imbalance issue. Empirically, we recommend a negative sample size of 3 to 5.

6 Conclusion

In this paper, we propose a relation-aware network with attention-based loss for FKGC tasks. We strategically select multiple negative samples instead of one and propose an attention-based loss to differentiate the importance of each negative sample. A dynamic relation-aware entity encoder is designed to learn a context-dependent entity representation. The experimental results demonstrate that RANA outperforms other SOTA baselines on two benchmark datasets.

Acknowledgement. The work is supported in part by NSF IIS-2007941.

References

1. Bordes, A., Glorot, X., Weston, J., Bengio, Y.: A semantic matching energy function for learning with multi-relational data. Machine Learning **94**(2), 233–259 (2014)
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NeurIPS **26**, 2787–2795 (2013)
3. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E., Mitchell, T.: Toward an architecture for never-ending language learning. In: AAAI. vol. 24, pp. 1306–1313 (2010)
4. Chen, M., Zhang, W., Zhang, W., Chen, Q., Chen, H.: Meta relational learning for few-shot link prediction in knowledge graphs. In: EMNLP-IJCNLP. pp. 4217–4226 (2019)
5. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML. pp. 1126–1135. PMLR (2017)
6. Kazemi, S.M., Poole, D.: Simple embedding for link prediction in knowledge graphs. In: NeurIPS **31** (2018)
7. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
8. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: ICML (2011)
9. Niu, G., Li, Y., Tang, C., Geng, R., Dai, J., Liu, Q., Wang, H., Sun, J., Huang, F., Si, L.: Relational learning with gated and attentive neighbor aggregator for few-shot knowledge graph completion. In: SIGIR. pp. 213–222 (2021)
10. Rong, Y., Huang, W., Xu, T., Huang, J.: Dropedge: Towards deep graph convolutional networks on node classification. In: ICLR (2019)
11. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: CVPR. pp. 815–823 (2015)
12. Sheng, J., Guo, S., Chen, Z., Yue, J., Wang, L., Liu, T., Xu, H.: Adaptive attentional network for few-shot knowledge graph completion. In: EMNLP. pp. 1681–1691 (2020)
13. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. In: In ICLR (2018)
14. Tang, Z., Pei, S., Zhang, Z., Zhu, Y., Zhuang, F., Hoehndorf, R., Zhang, X.: Positive-unlabeled learning with adversarial data augmentation for knowledge graph completion. In: IJCAI. pp. 1935–1942 (2022)
15. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: ICML. pp. 2071–2080. PMLR (2016)
16. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. CACM (2014)
17. Wu, H., Yin, J., Rajaratnam, B., Guo, J.: Hierarchical relational learning for few-shot knowledge graph completion. arXiv preprint arXiv:2209.01205 (2022)
18. Xiong, W., Yu, M., Chang, S., Guo, X., Wang, W.Y.: One-shot relational learning for knowledge graphs. In: EMNLP. pp. 1980–1990 (2018)
19. Yang, B., Yih, S.W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: ICLR (2015)
20. Yang, S., Tian, J., Zhang, H., Yan, J., He, H., Jin, Y.: Transms: Knowledge graph embedding for complex relations by multidirectional semantics. In: IJCAI. pp. 1935–1942 (2019)
21. Yang, Z., Ding, M., Zhou, C., Yang, H., Zhou, J., Tang, J.: Understanding negative sampling in graph representation learning. In: SIGKDD. pp. 1666–1676 (2020)
22. Zha, H., Chen, Z., Yan, X.: Inductive relation prediction by bert. In: AAAI (2022)
23. Zhang, C., Yao, H., Huang, C., Jiang, M., Li, Z., Chawla, N.V.: Few-shot knowledge graph completion. In: AAAI. vol. 34, pp. 3041–3048 (2020)
24. Zhou, K., Li, Y., Li, Q.: Distantly supervised named entity recognition via confidence-based multi-class positive and unlabeled learning. In: ACL. pp. 7198–7211 (2022)
25. Zhou, K., Qiao, Q., Li, Y., Li, Q.: Improving distantly supervised relation extraction by natural language inference. arXiv preprint arXiv:2208.00346 (2022)