Learning Inter-Entity Interaction for Few-Shot Knowledge Graph Completion

Yuling Li, Kui Yu*, Xiaoling Huang, Yuhong Zhang
Key Laboratory of Knowledge Engineering with Big Data, Ministry of Education, Hefei, China
School of Computer Science and Information Enginerring, Hefei University of Technology
lyl95@mail.hfut.edu.cn, {yukui,zhangyh}@hfut.edu.cn, hxl@chzu.edu.cn

Abstract

Few-shot knowledge graph completion (FKGC) aims to infer unknown fact triples of a relation using its few-shot reference entity pairs. Recent FKGC studies focus on learning semantic representations of entity pairs by separately encoding the neighborhoods of head and tail entities. Such practice, however, ignores the inter-entity interaction, resulting in low-discrimination representations for entity pairs, especially when these entity pairs are associated with 1-to-N, N-to-1, and N-to-N relations. To address this issue, this paper proposes a novel FKGC model, named Cross-Interaction Attention Network (CIAN) to investigate the *inter-entity interaction* between head and tail entities. Specifically, we first explore the interactions within entities by computing the attention between the task relation and each entity neighbor, and then model the interactions between head and tail entities by letting an entity to attend to the neighborhood of its paired entity. In this way, CIAN can figure out the relevant semantics between head and tail entities, thereby generating more discriminative representations for entity pairs. Extensive experiments on two public datasets show that CIAN outperforms several state-of-the-art methods. The source code is available at https://github.com/cjlyl/FKGC-CIAN.

1 Introduction

Knowledge graphs (KGs) like YAGO (Suchanek et al., 2007), Freebase (Bollacker et al., 2008), and Wikidata (Vrandečić and Krötzsch, 2014) have been successfully applied to various knowledge-driven applications, such as question answering (Saxena et al., 2020), semantic search (Xiong et al., 2017), and information retrieval (Liu et al., 2018). A knowledge graph consists of a large number of fact triples, where each triple is represented in the form of *(head entity, relation, tail entity)*. Although typical KGs are large in size, they still suffer from incompleteness. This motivates the research in knowledge graph completion (KGC), which aims to predict the missing elements in incomplete triples.

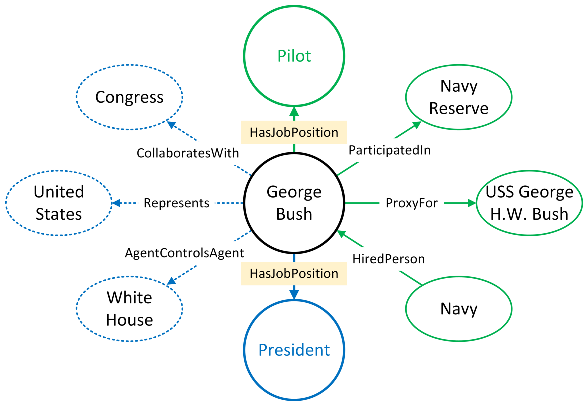

Figure 1: An example to illustrate the importance of the relevant attributes between head and tail entities. Intuitively, the political attribute (dashed blue) of head entity GeorgeBush could be more helpful for representing entity pair (GeorgeBush, President).

In literature, state-of-the-art KGC models are usually based on knowledge graph embeddings (KGE). The key idea behind KGE-based methods is to embed entities and relations into low-dimensional vector spaces and measure the plausibility of triples based on their embeddings (Bordes et al., 2013; Nickel et al., 2011). Despite their success, existing KGE-based models generally require sufficient training triples for all relations. However, in many real-world KGs, a large proportion of (few-shot) relations only contain a limited number of triples (Xiong et al., 2018; Chen et al., 2019; Sheng et al., 2020). The shortage of training triples hinders existing KGE-based models from achieving satisfactory performance on these few-shot relations.

To address this issue, few-shot knowledge graph

*Corresponding author: Kui Yu.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 7691 - 7700
December 7-11, 2022 ©2022 Association for Computational Linguistics

completion (FKGC) models have been proposed to predict a tail entity t in a query triple $(h, r, ?)$ given K support entity pairs about the task relation r. Their primary focus lies in learning semantic representations of entity pairs. Typically, most methods learn entity embeddings from their respective neighborhoods and then concatenate the embeddings of head and tail entities (Niu et al., 2021; Wang et al., 2021; Zhang et al., 2020). Such practice, however, treats each entity independently when modeling the semantic information of entity pairs, ignoring the interactions between head and tail entities. As a result, the generated representations of entity pairs are not sufficiently discriminative, especially in 1-to-N, N-to-1, and N-to-N relations where many different entity pairs involve common entities.

To solve this problem, we propose to investigate the *inter-entity interaction* between head and tail entities, which affects the importance of their semantic attributes to represent entity pairs. The basic idea behind this is that entity pairs are supposed to pay more attention to the relevant semantic attributes between head and tail entities. Figure 1 shows an example of the head entity `GeorgeBush` and its two entity pairs belonging to the task relation `HasJobPosition`. The left neighborhood (dashed blue) describes the political attribute of `GeorgeBush` as a former president of the US, while the right one (solid green) shows his soldier attribute as a navy pilot. Intuitively, the political attribute could be more helpful for representing (`GeorgeBush`, `President`), while the soldier attribute could contribute more to representing (`GeorgeBush`, `Pilot`). Hence, exploiting the interactions between head and tail entities is beneficial to characterize entity pairs.

To this end, we propose a **C**ross-**I**nteraction **A**ttention **N**etwork (CIAN) for few-shot knowledge graph completion. Our contributions can be summarized as follows:

- We investigate the *inter-entity interaction* in few-shot knowledge graph completion, which differs from previous paradigms by exploring the relevant semantic information between head and tail entities to represent corresponding entity pairs.

- We model the inter-entity interaction through two stages: the interactions within entities are first captured to extract task-relevant entity attributes, and then the interactions between entities are explored to discern the attributes of one entity related to its paired entity.

- We conduct extensive experiments on the NELL-One and Wiki-One datasets. Experimental results demonstrate that our method outperforms the state-of-the-art methods of few-shot KG completion with different few-shot sizes.

2 Related Work

A common approach for KGC is to represent entities and relations as low-dimensional vectors (a.k.a, knowledge graph embeddings, KGE) and then learn a well-designed score function to measure the plausibility of triples in the embedding space. Existing KGE-based models can be roughly classified into two groups: (1) translational distance approaches, which view relations as translation operations and define a distance-based score function accordingly (Bordes et al., 2013; Wang et al., 2014; Lin et al., 2015), and (2) semantic matching approaches, which establish similarity-based score functions and make predictions by matching latent semantics of entities and relations (Nickel et al., 2011; Trouillon et al., 2016; Vashishth et al., 2019; Schlichtkrull et al., 2018). These KGE-based models usually assume that there are sufficient training triples for all relations, and thus are limited in practical few-shot scenarios.

Recently, much research progress has been made in few-shot KG completion, which solves KGC for unseen relations with a handful of training samples, based on some prior knowledge gained from a large number of similar but different relations. Existing FKGC approaches can be divided into two main categories. (1) Metric learning-based methods: GMatching (Xiong et al., 2018) is the first study to solve the one-shot KG completion problem. It learns entity embeddings by encoding their direct neighbors and measures the similarities between the query and reference entity pairs by a multi-step matching processor. FSRL (Zhang et al., 2020) extends GMatching to a few-shot scenario through a recurrent autoencoder aggregation network and employs a heterogeneous neighbor encoder to improve the quality of entity embeddings. Unlike GMatching and FSRL that learn the identical representation for the same entity in various triples, FAAN (Sheng et al., 2020) allows for the dynamic properties of entities under different task relations and learns task-aware entity representations. In addition, P-INT (Xu et al., 2021) leverages the paths from head to tail entities to represent entity pairs and calculates the path interactions between the support and query entity pairs. (2) Meta learner-based methods: MetaR (Chen et al., 2019) generates relation-specific meta-information using the embeddings of head and tail entities and updates relation meta via gradient meta. GANA (Niu et al., 2021) learns relation-meta by incorporating the neighbor information of entities and devises a meta-learning-based TransH module to model complex relations.

Different from the aforementioned methods that treat each entity independently, our work explores the inter-entity interaction and leverages the relevant attributes between entities to represent entity pairs.

3 Problem Formulation

A KG can be represented as a set of triples $\mathcal{G} = \{(h, r, t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}\}$, where $\mathcal{E}$ and $\mathcal{R}$ are the entity set and the relation set, respectively. Given any two of three elements within a triple, the KG completion task aims to predict the remaining one. This study focuses on predicting t given $(h, r, ?)$ since our goal is to infer new facts for few-shot relations.

Following the standard setting in FKGC (Xiong et al., 2018; Chen et al., 2019), we have access to a background KG $\mathcal{G}'$, which is a subset of $\mathcal{G}$ with some relations of $\mathcal{R}$. The remaining relations in $\mathcal{R}$ are further divided into three disjoint task sets $\mathcal{R}_{train}$, $\mathcal{R}_{test}$ and $\mathcal{R}_{valid}$, which are used in the meta-training, meta-testing, and meta-validation phases, respectively.

To perform few-shot learning, we adopt the episodic paradigm (Vinyals et al., 2016) to imitate the real test conditions. Specifically, in each iteration of the meta-training phase, we sample a task relation r from $\mathcal{R}_{train}$, and construct its support set $\mathcal{S}_r = \{(h_i, t_i)\}_{i=1}^{K}$ and its query set $\mathcal{Q}_r = \{(h_j, t_j, \mathcal{C}_{h_j,r})\}_{j=1}^{B}$. Here, K and B denote the numbers of support and query triples, t_j is the ground-truth tail entity for query head entity h about relation r, and $\mathcal{C}_{h_j,r}$ is the set of the corresponding candidate entities in $\mathcal{G}$. The candidate entities are constructed based on the entity type constraint (Xiong et al., 2018; Toutanova et al., 2015). Our goal is to train a model using K support entity pairs in $\mathcal{S}_r$ such that the model can rank the ground-truth entity t_j higher than the corresponding candidate entities in $\mathcal{C}_{h_j,r}$. A few-shot KG completion task can be defined as follows:

Definition of Few-shot KG Completion. *Given a task relation r and its support set $\mathcal{S}_r = \{(h_i, r, t_i)\}_{i=1}^{K}$, one task is to predict the missing tail entities of the query triples, where few-shot size K is very small. The task is called K-shot KG completion.*

After sufficient training with $\mathcal{R}_{train}$, the learned model can be used for the meta-validation and meta-testing phases with $\mathcal{R}_{valid}$ and $\mathcal{R}_{test}$, which are defined in the same way as $\mathcal{R}_{train}$.

4 Methodology

The framework of the proposed CIAN is illustrated in Figure 2. CIAN includes the following stages: (1) modeling the inter-entity interaction by a task-aware attention module and an entity-pair-aware attention module to generate the semantic representations of entity pairs, and (2) matching support and query sets to compare the degree of semantic matching between the input queries and the given support entity pairs.

4.1 Modeling the Inter-entity Interaction

In this subsection, we aim to explore the *inter-entity interaction* between the head and tail entities and utilize their relevant semantics to represent the corresponding entity pairs. To achieve this goal, we take the following two steps: (1) First, we design a task-aware attention module to extract task-relevant semantic attributes for each entity. This step is motivated by FAAN (Sheng et al., 2020), which points out that entities should exhibit the semantic attributes that adapt to the task relations. (2) On the basis of the task-relevant attributes, we then explore the relevant semantic attributes between head and tail entities. This is realized by an entity-pair-aware attention module, which attempts to compute the attention of interaction between one entity and the neighbors of its paired entity. Figure 3 gives an example of modeling the inter-entity interaction. We detail each module in the following.

4.1.1 Task-aware Attention Module

Given a pair of entities (h, t) about the task relation r, the task-aware attention module takes as input each entity and its local neighborhood, and outputs its task-aware representation.

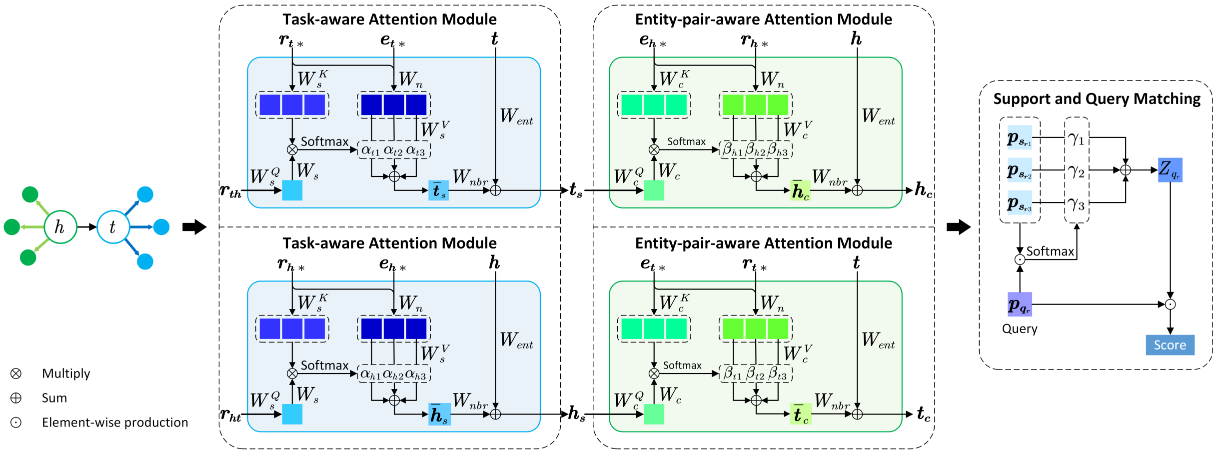

Figure 2: Illustration of CIAN model architecture.

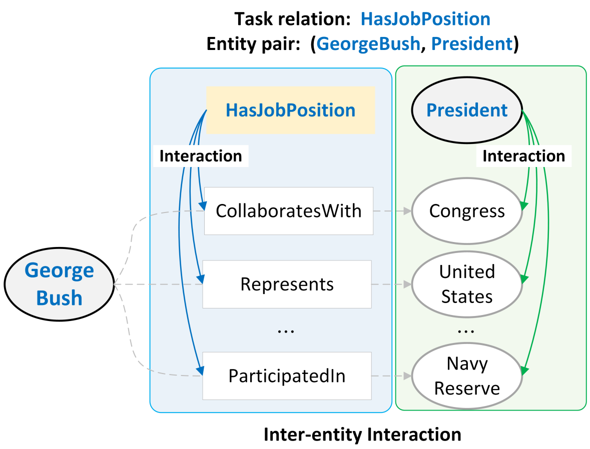

Figure 3: An example of modeling the inter-entity interaction through two interaction stages.

Take the head entity h as a target, and we denote its local neighborhood as $\mathcal{N}_h = \{(r_{hi}, e_{hi}) | (h, r_{hi}, e_{hi}) \in \mathcal{G}'\}$. We first leverage three linear transformations, parameterized by $W_s^Q, W_s^K, W_s^V \in \mathbb{R}^{d \times d}$, to project the task relation representation into the query, the neighboring relation representations into the keys, and the semantic representations of neighbors into the values, as follows:

$$\begin{aligned} Q_{ht}^s &= \boldsymbol{r_{ht}} W_s^Q, \\ K_{hi}^s &= \boldsymbol{r_{hi}} W_s^K, \\ V_{hi}^s &= \boldsymbol{n_{hi}} W_s^V, \end{aligned} \tag{1}$$

where

$$[\boldsymbol{r_{ht}} || \boldsymbol{r_{th}}] = \text{Bilinear}(\boldsymbol{h}, \boldsymbol{t}), \tag{2}$$

$$\boldsymbol{n_{hi}} = \text{ReLU}\big([\boldsymbol{r_{hi}} || \boldsymbol{e_{hi}}] W_n + b_n\big). \tag{3}$$

Here, $\boldsymbol{h}, \boldsymbol{t}, \boldsymbol{e_{hi}}, \boldsymbol{r_{hi}} \in \mathbb{R}^d$ are the embedding vectors of h, t, e_{hi}, r_{hi}, respectively; d is the embedding dimension; $||$ represents the concatenation operation; $W_n \in \mathbb{R}^{2d \times d}$ and $b_n \in \mathbb{R}^d$ are learnable parameters shared by all neighbors. In this paper, we adopt a bilinear function to model pairwise interactions between the head and tail entities (as shown in Eq. 2), and obtain the forward representation $\boldsymbol{r_{ht}} \in \mathbb{R}^d$ and the backward representation $\boldsymbol{r_{th}} \in \mathbb{R}^d$ about the task relation r. In addition, we apply a feed-forward neural network to learn the semantic representation $\boldsymbol{n_{hi}}$ of neighbor (r_{hi}, e_{hi}). Different from previous studies (Sheng et al., 2020; Zhang et al., 2020) that only use the information of neighboring entities to represent neighbor semantics, we compute the semantic representations of neighbors by combining the information of neighboring entities and neighboring relations, as shown in Eq. 3.

Next, we calculate the relevance between the neighboring relation and the task relation by a bilinear dot product, and then normalize the relevance scores across all neighbors using the softmax function, as follows:

$$\text{Att}_{s_{hi}} = Q_{ht}^s W_s {K_{hi}^s}^T, \tag{4}$$

$$\alpha_{hi} = \frac{\exp(\text{Att}_{s_{hi}})}{\sum_{(r_{hj}, e_{hj}) \in \mathcal{N}_h} \exp(\text{Att}_{s_{hj}})}, \tag{5}$$

where $W_s \in \mathbb{R}^{d \times d}$ is a trainable weight vector that captures interactions between relations; α_{hi} represents the task-relevant attention weight of neighbor (r_{hi}, e_{hi}). Afterward, we obtain the task-aware neighbor representation $\bar{\boldsymbol{h}}_{\boldsymbol{s}}$ about h by considering its diverse roles:

$$\bar{\boldsymbol{h}}_{\boldsymbol{s}} = \sum_{(r_{hi}, e_{hi}) \in \mathcal{N}_h} \alpha_{hi} V_{hi}^s. \tag{6}$$

In this way, entity neighbors that are more relevant to the task relation will be assigned higher weights, thus capturing more valuable neighbor information.

Finally, we incorporate the weighted neighbor representation $\bar{\boldsymbol{h}}_{\boldsymbol{s}}$ into the entity embedding $\boldsymbol{h}$ to generate the task-aware entity representation:

$$\boldsymbol{h_s} = \mathrm{ReLU}(\boldsymbol{h}W_{ent} + \bar{\boldsymbol{h}}_{\boldsymbol{s}}W_{nbr}), \tag{7}$$

where W_{ent}, $W_{nbr} \in \mathbb{R}^{d\times d}$ are two trainable weight vectors.

Likewise, we also perform the above procedures on tail entity t and its local neighborhood $\mathcal{N}_t$, and obtain its task-aware entity representation $\boldsymbol{t_s}$.

4.1.2 Entity-pair-aware Attention Module

Given a pair of entities, the entity-pair-aware attention module aims to identify the relevant semantics between head and tail entities by letting one entity attend to each neighbor of its paired entity. The generated entity-pair-aware representations are then coupled together to form the semantic representation of the corresponding entity pair.

Take the head entity h as a target. The entity-pair-aware attention module takes as input $\{\boldsymbol{h}, \mathcal{N}_h, \boldsymbol{t_s}\}$ and outputs the entity-pair-aware representation of h. In specific, for each neighbor $(r_{hi}, e_{hi}) \in \mathcal{N}_h$, we transform the task-aware representation of t into the query, the neighboring entity representation $\boldsymbol{e_{hi}}$ into the keys, and the neighbor representation $\boldsymbol{n_{hi}}$ into the values with three linear transformations, $W_c^Q, W_c^K, W_c^V \in \mathbb{R}^{d\times d}$, respectively, as follows:

$$\begin{aligned} Q_t^c &= \boldsymbol{t_s}W_c^Q, \\ K_{hi}^c &= \boldsymbol{e_{hi}}W_c^K, \\ V_{hi}^c &= \boldsymbol{n_{hi}}W_c^V. \end{aligned} \tag{8}$$

Then, we compute the relevance of each neighbor with respect to t:

$$\mathrm{Att}_{c_{hi}} = Q_t^c W_c {K_{hi}^c}^T, \tag{9}$$

where $W_c \in \mathbb{R}^{d\times d}$ is a trainable weight vector that captures the interactions between entities. Afterward, we compute the entity-pair-aware neighbor representation $\bar{\boldsymbol{h}}_{\boldsymbol{c}}$ for h:

$$\begin{aligned} \bar{\boldsymbol{h}}_{\boldsymbol{c}} &= \sum_{(r_{hi},e_{hi})\in\mathcal{N}_h} \beta_{hi}V_{hi}^c, \\ \beta_{hi} &= \frac{\exp(\mathrm{Att}_{c_{hi}})}{\sum\limits_{(r_{hj},e_{hj})\in\mathcal{N}_h} \exp(\mathrm{Att}_{c_{hj}})}, \end{aligned} \tag{10}$$

where β_{hi} denotes the entity-pair-aware attention weight of neighbor (r_{hi}, e_{hi}). Higher values of β_{hi} are indicative of stronger relevance between the i-th entity neighbors of h and the paired entity t.

Finally, we generate the entity-pair-aware representation of h by combining its entity embedding $\boldsymbol{h}$ and its neighbor representation:

$$\boldsymbol{h_c} = \mathrm{ReLU}(\boldsymbol{h}W_{ent} + \bar{\boldsymbol{h}}_{\boldsymbol{c}}W_{nbr}). \tag{11}$$

Analogously, for tail entity t, the attention mechanism takes as input $\{\boldsymbol{t}, \mathcal{N}_t, \boldsymbol{h_s}\}$ and generates its entity-pair-aware representation $\boldsymbol{t_c}$. Based on the generated representations $\boldsymbol{h_c}$ and $\boldsymbol{t_c}$, we compute the semantic representation $\boldsymbol{p_{(h,t)}}$ of entity pair (h, t) by a position-wise forward feed neural network:

$$\boldsymbol{p_{(h,t)}} = \mathrm{ReLU}\big([\boldsymbol{h_c}||\boldsymbol{t_c}]W_{p1} + b_p\big)W_{p2}, \tag{12}$$

where $W_{p1}, W_{p2} \in \mathbb{R}^{2d\times 2d}$ and $b_p \in \mathbb{R}^{2d}$ are trainable matrix vectors.

4.2 Matching Support and Query Sets

Based on the generated semantic representations of entity pairs, we are going to learn a general representation from the representations of the reference entity pairs in the support set, i.e., the prototype of the few-shot relation. Due to the scarcity of reference triples, one reference triple far from other references in the vector space will cause a huge deviation of the corresponding prototype (Gao et al., 2019). Inspired by (Sheng et al., 2020; Gao et al., 2019), we introduce an attention mechanism to select more informative references and denoise the noisy references during training.

In specific, given a query q_r in the query set $\mathcal{Q}_r$, the prototype vector Z_{q_r} of few-shot relation r can be obtained as follows:

$$Z_{q_r} = \sum_{s_{rk}\in\mathcal{S}_r} \gamma_k \boldsymbol{p_{s_{rk}}}, \tag{13}$$

$$\gamma_k = \frac{\exp(\boldsymbol{p_{q_r}} \odot \boldsymbol{p_{s_{rk}}})}{\sum_{s_{ri}\in\mathcal{S}_r} \exp(\boldsymbol{p_{q_r}} \odot \boldsymbol{p_{s_{ri}}})}. \tag{14}$$

Here, $s_{rk} \triangleq (h_k, r, t_k) \in \mathcal{S}_r$ denotes the k-th triple in the support set $\mathcal{S}_r$; $\odot$ denotes element-wise production. As such, the prototype Z_{q_r} focuses more on the query-relevant references and thus reduces the impact of noisy references.

Dataset	# Ent	# Triple	# Rel	# Task
NELL-One	68,545	181,109	358	67
Wiki-One	4,838,244	5,859,240	822	183

Table 1: Statistics of datasets. # Ent, # Triple, # Rel and # Task represent the number of entities, triples, relations, and task relations, respectively.

To predict the plausibility of q_r , we calculate the similarity score between the query representation of q_r and the prototype vector of r, as follows:

$$\phi(q_r, S_r) = \boldsymbol{p_{q_r}} \odot Z_{q_r}. \tag{15}$$

Higher values of $\phi(\cdot)$ are indicative of more reasonable triples.

4.3 Loss Function and Model Training

Given a task $r \in \mathcal{R}_{train}$ and its corresponding triples, we randomly sample K entity pairs to construct the support set $\mathcal{S}_r$, and a batch of entity pairs to construct the positive query set $\mathcal{Q}_r$. For each entity pair in $\mathcal{Q}_r$, we pollute its tail entity and construct a set of negative query entity pairs $\mathcal{Q}_r^- = \{(h_q, t_q^-)|(h_q, t_q) \in \mathcal{Q}_r\}$, where $t_q^- \in \{\mathcal{C}_{h_q,r} \setminus t_q\}$. Then we use a margin-based scoring function to ensure that a positive query in $\mathcal{Q}_r$ has a higher similarity score than a negative query in $\mathcal{Q}_r^-$, as follows:

$$\mathcal{L} = \sum_r \sum_{q_r \in \mathcal{Q}_r} \sum_{q_r^- \in \mathcal{Q}_r^-} [\gamma + \phi(q_r^-, S_r) - \phi(q_r, S_r)]_+, \tag{16}$$

where $[x]_+ = \max(0, x)$ is the hinge loss, and $\gamma > 0$ is a margin hyperparameter. We adopt a batch sampling-based meta-training procedure (Vinyals et al., 2016) to minimize $\mathcal{L}$ and optimize model parameters.

5 Experiments

In this section, we conduct extensive experiments to evaluate the performance of the proposed CIAN and verify the effectiveness of the key components in CIAN.

5.1 Experimental Settings

Datasets. In our experiments, we choose two widely used benchmark datasets, namely NELL-One and Wiki-One (Xiong et al., 2018), for few-shot KG completion. In both datasets, relations with less than 500 but more than 50 triples are selected as few-shot relations, and the remaining relations along with their triples constitute the background knowledge graphs. With this setting, there are 67 and 183 few-shot relations on NELL-One and Wiki-One, respectively. Following the original settings (Xiong et al., 2018), we split the training/test/validation relations as 51/11/5 and 133/34/16 on NELL-One and Wiki-One, respectively. The detailed statistics of the two datasets are summarized in Table 1.

Evaluation Metrics. We evaluate the model performance using two common metrics: MRR and Hits@N. MRR is the average of the reciprocal ranks of the correct entities. Hits@N is the proportion of correct entities ranked in the top N, with $N = 1, 5, 10$. Higher values of MRR or Hits@N are indicative of the better performance of KG completion.

Comparison Methods. We choose the following 5 state-of-the-art models[1] as the baselines to compare with CIAN: GMatching (Xiong et al., 2018), MetaR (Chen et al., 2019), FSRL (Zhang et al., 2020), FAAN (Sheng et al., 2020), and GANA (Niu et al., 2021) (cf. Section 2 for model details). All the above methods learn entity embeddings by independently encoding their neighborhoods. Traditional KGE-based models such as TransE, DisMult, and ComplEx have been shown to perform significantly worse than the aforementioned few-shot baselines (Sheng et al., 2020; Niu et al., 2021). Due to space limitations, we omit the comparisons with these traditional methods.

Implementation Details. We initialize the entity and relation embeddings with the pre-trained TransE embeddings provided by Xiong et al. (2018), and further fine-tune these embeddings during training. On both datasets, we set the maximum number of neighbors as 100, and fix the margin γ to 5.0. We use the Adam optimizer (Kingma and Ba, 2015) as the optimizer with an initial learning rate of $8e^{-5}$ for NELL-One and $2e^{-4}$ for Wiki-One, respectively. Following (Sheng et al., 2020), we evaluate our model on the validation set at every 10k training steps, and save the best model when MRR reaches the highest value within 300k steps. We select the optimal hyperparameters of our method by grid search on the validation set.

[1]For the sake of fairness, we do not compare our method with P-INT (Xu et al., 2021) since its results are obtained by discarding disconnected entity pairs.

NELL-One	MRR 3-shot	MRR 5-shot	Hits@10 3-shot	Hits@10 5-shot	Hits@5 3-shot	Hits@5 5-shot	Hits@1 3-shot	Hits@1 5-shot
GMatching (Xiong et al., 2018)	-	.176‡	-	.294‡	-	.233‡	-	.113‡
MetaR (In-Train) (Chen et al., 2019)	.245†	.261	.456†	.437	.360†	.350	.144†	.168
MetaR (Pre-Train) (Chen et al., 2019)	.210†	.209	.386†	.355	.311†	.280	.119†	.141
FSRL (Zhang et al., 2020)	.219†	.195†	.383†	.359†	.296†	.279†	.139†	.108†
FAAN (Sheng et al., 2020)	.247†	.279	.369†	.428	.309†	.364	.183†	.200
GANA (Niu et al., 2021)	.322	.344	**.510**	.517	**.432**	.437	.225	.246
CIAN	**.344**	**.376**	.484	**.527**	.417	**.453**	**.266**	**.298**
Wiki-One	**3-shot**	**5-shot**	**3-shot**	**5-shot**	**3-shot**	**5-shot**	**3-shot**	**5-shot**
GMatching (Xiong et al., 2018)	-	.263‡	-	.387‡	-	.337‡	-	.197‡
MetaR (In-Train) (Chen et al., 2019)	.210†	.221	.299†	.302	.249†	.264	.165†	.178
MetaR (Pre-Train) (Chen et al., 2019)	.317†	.323	.432†	.418	.379†	.385	.261†	.270
FSRL (Zhang et al., 2020)	.102†	.113†	.200†	.236†	.131†	.135†	.050†	.056†
FAAN (Sheng et al., 2020)	.298†	.341	.435†	.463	.368†	.395	.228†	.281
GANA (Niu et al., 2021)	.331	.351	.425	.446	.389	.407	.283	.299
CIAN	**.358**	**.383**	**.492**	**.505**	**.438**	**.453**	**.284**	**.318**

Table 2: Evaluation results of all methods on NELL-One and Wiki-One in terms of MRR and Hits@{1, 5, 10}. ‡Results come from (Sheng et al., 2020). †Results were obtained with the official implementation from the authors. The remaining results were reported in the original papers. For each metric, bold numbers mark the best results.

5.2 Overall Comparison with Baselines

Table 2 reports the performance of all models on the NELL-One and Wiki-One datasets. From the table, we can observe that our CIAN significantly outperforms all baselines on both datasets. Specifically, for 5-shot link prediction, CIAN achieves relative performance improvements of 3.2% / 1.0% / 1.6% / 5.2% in MRR / Hits@10 / Hits@5 / Hits@1 on NELL-One, and relative improvements of 3.2% / 4.2% / 4.6% / 1.9% on Wiki-One, compared to the best performing baseline. For 3-shot link prediction, CIAN achieves relative improvements of 2.7% / 5.7% / 4.9% / 0.1% in MRR / Hits@10 / Hits@5 / Hits@1 on Wiki-One over the best performing baseline. These results demonstrate that exploiting the interactions between head and tail entities can indeed improve the performance of few-shot KG completion.

To further investigate the effectiveness of our model under different few-shot sizes, we compare the performance of all models on NELL-One in various settings of K. The results are shown in Figure 4. It can be observed that:

(1) CIAN consistently outperforms all baselines by a large margin under different K, further proving the superiority of our model in the few-shot scenario.

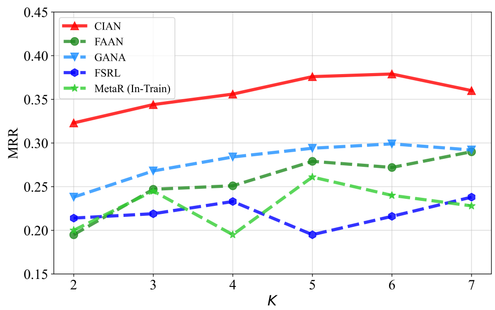

Figure 4: Impact of few-shot size K on NELL-One.

(2) With the increment of K, the performance of all models does not continuously increase, suggesting that larger reference sets do not always lead to better performance of FKGC models. Even so, CIAN still gets relatively stable improvements compared to most baselines such as FSRL and MetaR (In-Train).

5.3 Comparison with Variants

To evaluate the contributions of each component in the proposed CIAN, we compare CIAN with its three variant models. The results are displayed in Table 3. We observe that the performance drops significantly when the task-aware attention module is removed (-TAM). This demonstrates that suppressing task-irrelevant entity attributes can effectively

Configuration	MRR	Hits@10	Hits@5	Hits@1
Full model	**.376**	**.527**	**.453**	**.298**
-TAM	.307	.494	.408	.212
-EPAM	.211	.494	.408	.212
-Eq.2	.371	.484	.404	.223

Table 3: Results of CIAN and its variants in 5-shot link prediction on NELL-One. "-TAM" and "-EPAM" represent CIAN without task-aware attention module, and without entity-pair-aware attention module, respectively. "-Eq.2" means that we replace Eq.2 with the equation $\boldsymbol{r} = \boldsymbol{t} - \boldsymbol{h}$ used in FAAN.

improve the model performance. We also notice that removing the entity-pair-aware attention module (-EPAM) will cause a remarkable performance drop, and it performs worse than -TAM. This implies that modeling the interactions between entities is more critical than encoding each entity's neighborhood individually. In addition, we replace the computation method of task relations in Eq. 2 with the modeling method of task relation embeddings used in FAAN. It can be seen that the performance drops, indicating the superiority of our method to model relationship semantics between entities.

5.4 Comparison on Different Relations

To investigate the effectiveness of our model for different categories of relations, we further analyze the model performance of each task relation on the NELL testing data. Table 4 displays the results of CIAN and two best baselines: FAAN, and GANA. We can observe that CIAN has better performance on almost all relations. Further, the improvement is more significant in N-to-1, and 1-to-N relations than in 1-to-1 relations[2]. This phenomenon illustrates that utilizing the relevant semantics between head and tail entities to represent entity pairs can help model complex relations.

In addition, we visualize the 2D embeddings of positive and negative candidate entity pairs for ID#1 relation (`ProducedBy`) in Figure 5. From the figure, we can find that the representations of positive and negative entity pairs learned by our model can be clearly distinguished, which demonstrates that our model can generate more discriminative representations of entity pairs.

[2]We classify relation categories based on the definition of complex relations from (Wang et al., 2014)

ID	Category	CIAN	FAAN	GANA
1	N-to-1	**.632**	.488	.577
2		**.478**	.428	.022
3		**.607**	.597	.428
4		**.980**	.965	.972
5	1-to-N	**.350**	.073	.255
6		**.158**	.073	.139
7		**.167**	.139	.132
8		**.589**	.577	.463
9	1-to-1	**.270**	.220	.246
10		**.429**	.391	.246
11		.012	.008	**.143**

Table 4: Results of CIAN, FAAN and GANA for each relation (RID) in NELL testing data.

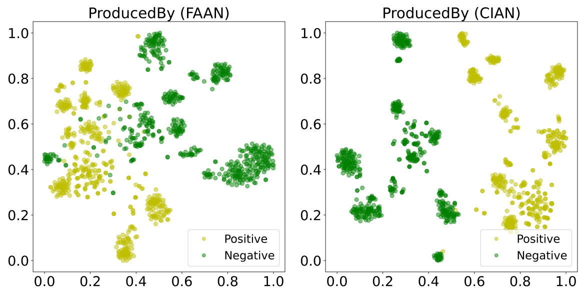

Figure 5: Embedding visualization of entity pairs about relation `ProducedBy`.

6 Conclusion

In this paper, we investigate the *inter-entity interaction* and propose a cross-interaction attention network for few-shot KG completion. Specifically, a task-aware attention module is designed to extract task-relevant entity attributes, and an entity-pair-aware attention module is developed to identify the relevant semantic attributes between head and tail entities. Experiments on two public datasets demonstrate that our model outperforms state-of-the-art models with different few-shot sizes. In the future, we will study how to make use of the side information such as textual descriptions of entities to solve the few-shot KG completion task.

Acknowledgements

This work is supported by the National Key Research and Development Program of China (under grant 2020AAA0106100), National Natural Science Foundation of China (under Grant 61876206), Open Project Foundation of Intelligent Information Processing Key Laboratory of Shanxi Province (under grant CICIP2020003), and the Key Project of the Natural Science Foundation of Educational Commission of Anhui Province (under grant KJ2021A1065).

Limitations

Our method has the following limitations:

- The performance of our model may degrade when there are many long-tailed entities with few direct neighbors. This is because sparse neighborhoods are not beneficial to model the inter-entity interaction.

- Since our method focuses on exploiting the interactions between entity neighborhoods, it requires a background graph to provide the neighbors of entities.

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, pages 2787–2795.

Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. 2019. Meta relational learning for few-shot link prediction in knowledge graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 4217–4226.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6407–6414.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*, pages 2181–2187.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2018. Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 2395–2405.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Icml*, pages 809–816.

Guanglin Niu, Yang Li, Chengguang Tang, Ruiying Geng, Jian Dai, Qiao Liu, Hao Wang, Jian Sun, Fei Huang, and Luo Si. 2021. Relational learning with gated and attentive neighbor aggregator for few-shot knowledge graph completion. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 213–222.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Jiawei Sheng, Shu Guo, Zhenyu Chen, Juwei Yue, Lihong Wang, Tingwen Liu, and Hongbo Xu. 2020. Adaptive attentional network for few-shot knowledge graph completion. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 1681–1691.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1499–1509.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2019. Composition-based multi-relational graph convolutional networks. In *International Conference on Learning Representations*, pages 1–15.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, pages 3630–3638.

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

Song Wang, Xiao Huang, Chen Chen, Liang Wu, and Jundong Li. 2021. Reform: Error-aware few-shot knowledge graph completion. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1979–1988.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1112–1119.

Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th international conference on world wide web*, pages 1271–1279.

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2018. One-shot relational learning for knowledge graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1980–1990.

Jingwen Xu, Jing Zhang, Xirui Ke, Yuxiao Dong, Hong Chen, Cuiping Li, and Yongbin Liu. 2021. P-int: A path-based interaction model for few-shot knowledge graph completion. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 385–394.

Chuxu Zhang, Huaxiu Yao, Chao Huang, Meng Jiang, Zhenhui Li, and Nitesh V Chawla. 2020. Few-shot knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3041–3048.