Adaptive Attentional Network for Few-Shot Knowledge Graph Completion

Jiawei Sheng[1,2], **Shu Guo**[3], **Zhenyu Chen**[4], **Juwei Yue**[4], **Lihong Wang**[3]*, **Tingwen Liu**[1,2] and **Hongbo Xu**[1,2]
[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] National Computer Network Emergency Response Technical Team/Coordination Center of China
[4] Beijing Advanced Innovation Center of Big Data and Brain Computing, Beihang University
shengjiawei@iie.ac.cn, guoshu@cert.org.cn, wlh@isc.org.cn

Abstract

Few-shot Knowledge Graph (KG) completion is a focus of current research, where each task aims at querying unseen facts of a relation given its few-shot reference entity pairs. Recent attempts solve this problem by learning static representations of entities and references, ignoring their dynamic properties, i.e., entities may exhibit diverse roles within task relations, and references may make different contributions to queries. This work proposes an adaptive attentional network for few-shot KG completion by learning adaptive entity and reference representations. Specifically, entities are modeled by an adaptive neighbor encoder to discern their task-oriented roles, while references are modeled by an adaptive query-aware aggregator to differentiate their contributions. Through the attention mechanism, both entities and references can capture their fine-grained semantic meanings, and thus render more expressive representations. This will be more predictive for knowledge acquisition in the few-shot scenario. Evaluation in link prediction on two public datasets shows that our approach achieves new state-of-the-art results with different few-shot sizes. The source code is available at https://github.com/JiaweiSheng/FAAN.

1 Introduction

Knowledge Graphs (KGs) like Freebase (Bollacker et al., 2008), NELL (Carlson et al., 2010) and Wikidata (Vrandecic and Krötzsch, 2014) are extremely useful resources for NLP tasks, such as information extraction (Liu et al., 2018), machine reading (Yang and Mitchell, 2017), and relation extraction (Ren et al., 2017). A typical KG is a multi-relational graph, represented as triples of the form (h, r, t), indicating that two entities are connected by relation r. Although a KG contains a great number of triples, it is also known to suffer from incompleteness problem. KG completion, which aims at automatically inferring missing facts by examining existing ones, has thus attracted broad attention. A promising approach, namely KG embedding, has been proposed and successfully applied to this task. The key idea is to embed KG components, including entities and relations, into a continuous vector space and make predictions with their embeddings.

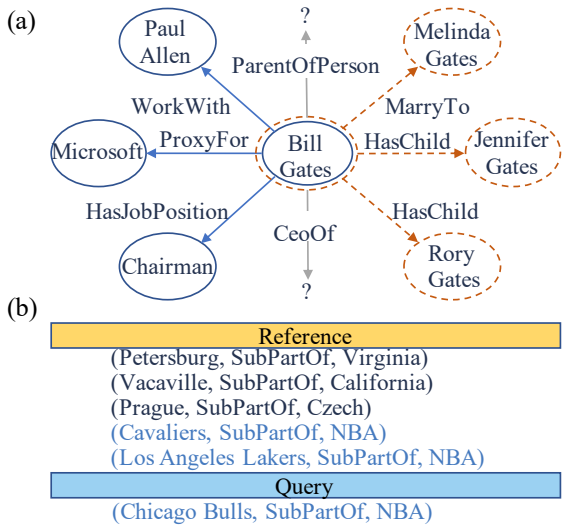

Figure 1: Illustration of dynamic properties in few-shot KG completion: (a) An entity has diverse roles in different tasks; and (b) References show distinct contributions to a particular query.

Current KG embedding methods mostly require sufficient training triples for all relations to learn expressive representations (*i.e.,* embeddings). In real KGs, a large portion of KG relations is actually long-tail, having only a limited (few-shot) number of relational triples (Xiong et al., 2018). This may lead to low performance of embedding models on KG completion for those long-tail relations.

Recently, several studies (Chen et al., 2019; Xiong et al., 2018; Zhang et al., 2020) have pro-

*Corresponding author: Lihong Wang.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 1681–1691,
November 16–20, 2020. ©2020 Association for Computational Linguistics

posed to address the few-shot issue of KG completion, where one task is to predict tail entity t in a query $(h, r, ?)$ given only a few entity pairs of the task relation r. These known few-shot entity pairs associated with r are called references. To improve semantical representations of the references, Xiong et al. (2018) and Zhang et al. (2020) devise modules to enhance entity embeddings with their local graph neighbors. The former simply assumes that all neighbors contribute equally to the entity embedding, and in this way the neighbors are always weighted identically. The latter develops the idea by employing an attention mechanism to assign different weights to neighbors, but the weights do not change throughout all task relations. Therefore, both works assign static weights to neighbors, leading to static entity representations when involved in different task relations. We argue that entity neighbors could have varied impacts associated with different task relations. Figure 1(a) gives an example of head entity `BillGates` associated with two task relations. The left neighbors show his business role, while the right ones show his family role, which reveals quite different meanings. Intuitively, the task relation `CeoOf` is supposed to pay more attention to the business role of entity `BillGates` than the other one.

In addition, task relations can be polysemous, also showing different meanings when involved in different entity pairs. Therefore, the reference triples could also make different contributions to a particular query. Take a task relation `SubPartOf` as an example. As shown in Figure 1(b), `SubPartOf` associates with different meanings, *e.g.*, organization-related as $(\texttt{Cavaliers}, \texttt{SubPartOf}, \texttt{NBA})$ and location-related as $(\texttt{Petersburg}, \texttt{SubPartOf}, \texttt{Virginia})$. Obviously, for query $(\texttt{ChicagoBulls}, \texttt{SubPartOf}, \texttt{?})$, referring to the organization-related references would be more beneficial.

To address the above issues, we propose an Adaptive Attentional Network for Few-Shot KG completion (FAAN), a novel paradigm that takes dynamic properties into account for both entities and references. Specifically, given a task relation with its reference/query triples, FAAN proposes an adaptive attentional neighbor encoder to model entity representations with one-hop entity neighbors. Unlike the previous neighbor encoder with a fixed attention map in (Zhang et al., 2020), we allow attention scores dynamically adaptive to the task relation under the translation assumption. This will capture the diverse roles of entities through varied impacts of neighbors. Given the enhanced entity representations, FAAN further adopts a stack of Transformer blocks for reference/query triples to capture multi-meanings of the task relation. Then, FAAN obtains a general reference representation by adaptively aggregating the references, further differentiating their contributions to different queries. As such, both entities and references can capture their fine-grained meanings, and render richer representations to be more predictive for knowledge acquisition in the few-shot scenario.

The contributions of this paper are three-fold:

(1) We propose the notion of dynamic properties in few-shot KG completion, which differs from previous paradigms by studying the dynamic nature of entities and references in the few-shot scenario.

(2) We devise a novel adaptive attentional network FAAN to learn dynamic representations. An adaptive neighbor encoder is used to adapt entity representations to different tasks. A Transformer encoder and an attention-based aggregator are used to adapt reference representations to different queries.

(3) We evaluate FAAN in few-shot link prediction on benchmark KGs of NELL and Wikidata. Experimental results reveal that FAAN could achieve new state-of-the-art results with different few-shot sizes.

2 Related Work

Recent years have seen increasing interest in learning representations for entities and relations in KGs, *a.k.a* KG embedding. Various methods have been devised, and roughly fall into three groups: 1) translation-based models which interpret relations as translating operations between head-tail entity pairs (Bordes et al., 2013; Yang et al., 2019); 2) simple semantic matching models which compute composite representations over entities and relations using linear mapping operations (Yang et al., 2015; Trouillon et al., 2016; Liu et al., 2017; Sun et al., 2019); and 3) (deep) neural network models which obtain composite representations using more complex operations (Schlichtkrull et al., 2018; Dettmers et al., 2018). Please refer to (Nickel et al., 2016; Wang et al., 2017; Ji et al., 2020) for a thorough review of KG embedding techniques. Traditional embedding models always require sufficient training triples for all relations, thus are

limited when solving the few-shot problem.

Previous few-shot learning studies mainly focus on computer vision (Sung et al., 2018), imitation learning (Duan et al., 2017) and sentiment analysis (Li et al., 2019). Recent attempts (Xiong et al., 2018; Chen et al., 2019; Zhang et al., 2020) tried to perform few-shot relational learning for long-tail relations. Xiong et al. (2018) proposed a matching network GMatching, which is the first research on one-shot learning for KGs as far as we know. GMatching exploits a neighbor encoder to enhance entity embeddings from their one-hop neighbors, and uses a LSTM matching processor to perform a multi-step matching by a LSTM block. FSRL (Zhang et al., 2020) extends GMatching to few-shot cases, further capturing local graph structures with an attention mechanism. Chen et al. (2019) proposed a novel meta relational learning framework MetaR by extracting and transferring shared knowledge across tasks from a few existing facts to incomplete ones. However, previous studies learn static representations of entities or references, ignoring their dynamic properties. This work attempts to learn dynamic entity and reference representations by an adaptive attentional network.

Dynamic properties have also been explored in other contexts outside few-shot relational learning. Ji et al. (2015); Wang et al. (2019) performed KG completion by learning dynamic entity and relation representations, but their methods are specially devised for traditional KG completion. Lu et al. (2017) adopted an adaptive attentional model for image captioning. Luo et al. (2019) tried to model dynamic user preference using a recurrent network with adaptive attention for the sequential recommendation. All these studies demonstrate the capability of modeling dynamic properties to enhance learning algorithms.

3 Background

Gonsider a KG $\mathcal{G}$ containing a set of triples $\mathcal{T} = \{(h, r, t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}\}$, where $\mathcal{E}$ and $\mathcal{R}$ denotes the entity set and relation set, respectively. This work focuses on a challenging link prediction scenario, *i.e.*, few-shot KG completion. We follow the standard definition of this task (Zhang et al., 2020):

Definition 1 (Few-shot KG Completion) *Given a relation $r \in \mathcal{R}$ and its reference set $\mathcal{S}_r = \{(h_k, t_k)|(h_k, r, t_k) \in \mathcal{T}\}$, one task is to complete triple (h, r, t) with tail entity $t \in \mathcal{E}$ missing, i.e., to predict t from a candidate entity set $\mathcal{C}$ given $(h, r, ?)$. When $|\mathcal{S}_r| = K$ and K is very small, the task is called K-shot KG completion.*

For this task, the goal of a few-shot learning method is to rank the true tail entity higher than false candidate entities, given few-shot reference entity pairs $\mathcal{S}_r$. To imitate such a link prediction task, each training task corresponds to a relation $r \in \mathcal{R}$ with its own reference/query entity pairs, *i.e.*, $\mathcal{D}_r = \{\mathcal{S}_r, \mathcal{Q}_r\}$, where $\mathcal{S}_r$ only consists of K-shot reference entity pairs (h_k, t_k). Additionally, $\mathcal{Q}_r = \{(h_m, t_m/\mathcal{C}_{h_m,r})\}$ contains all queries with ground-truth tail entity t_m and the corresponding candidates $\mathcal{C}_{h_m,r}$, where each candidate is an entity in $\mathcal{E}$ selected based on the entity type constraint (Xiong et al., 2018). The few-shot learning method thus could be trained on the task set by ranking the candidates in $\mathcal{C}_{h_m,r}$ given the query $(h_m, r, ?)$ and its references $\mathcal{S}_r$. All tasks in training form the *meta-training* set, denoted as $\mathcal{T}_{mtr} = \{\mathcal{D}_r\}$. Here, we only consider a closed set of entities appearing in $\mathcal{E}$.

After suffcient training with meta-training set, the learned model can be used to predict facts of new relation $r' \in \mathcal{R}'$ in testing. The relations used for testing are $unseen$ from meta-training, *i.e.*, $\mathcal{R}' \cup \mathcal{R} = \phi$. Each testing relation r' also has its own few-shot references and queries, *i.e.*, $\mathcal{D}_{r'} = \{\mathcal{S}_{r'}, \mathcal{Q}_{r'}\}$, defined in the same way as in meta-training. All tasks in testing form the *meta-testing* set, denoted as $\mathcal{T}_{mte} = \{\mathcal{D}_{r'}\}$. In addition, we also suppose that the model has access to a background KG $\mathcal{G}'$, which is a subset of $\mathcal{G}$ with all the relations excluded from $\mathcal{T}_{mtr}$ and $\mathcal{T}_{mte}$.

4 Our Approach

This section introduces our approach FAAN. Given a meta-training set $\mathcal{T}_{mtr}$, the purpose of FAAN is to learn a metric function for predictions by comparing the input query to the given references. To achieve this goal, FAAN consists of three major parts: (1) Adaptive neighbor encoder to learn adaptive entity representations; (2) Transformer encoder to learn relational representations for entity pairs; (3) Adaptive matching processor to compare the query to the given references. Finally, we present the detailed training objective of our model. Figure 2 shows the overall framework of FAAN for a task relation `CeoOf`.

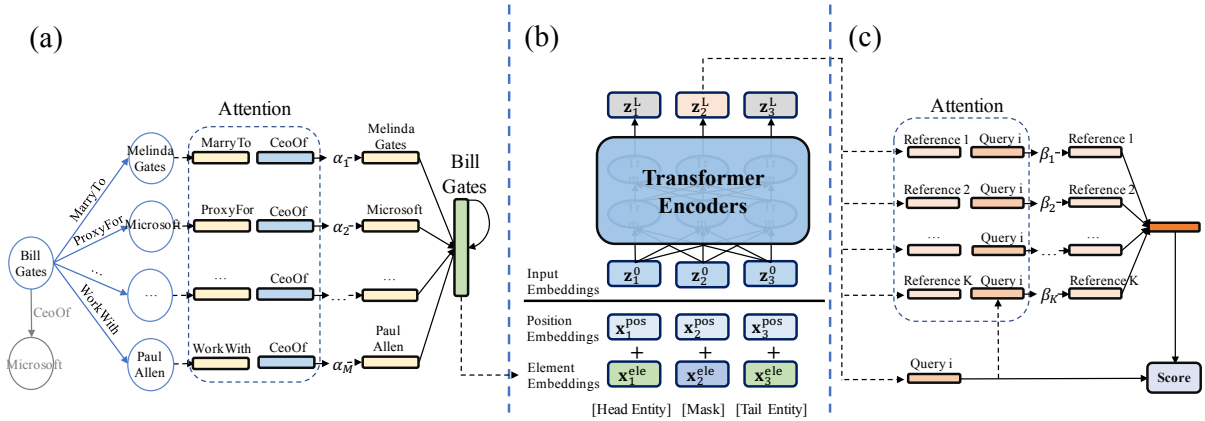

Figure 2: The framework of FAAN: (a) Adaptive neighbor encoder for entities; (b) Transformer encoder for entity pairs; (c) Adaptive matching processor to match K-shot references and the query.

4.1 Adaptive Neighbor Encoder for Entities

Previous works on embeddings (Schlichtkrull et al., 2018; Shang et al., 2019) have demonstrated that explicitly modeling graph contexts benefits KG completion. Recent few-shot relational learning methods encode one-hop neighbors to enhance entity embeddings with equal or fixed attentions (Xiong et al., 2018; Zhang et al., 2020), ignoring the dynamic properties of entities. To tackle this issue, we devise an adaptive neighbor encoder for entities discerning their entity roles associated with task relations. Specifically, we are given a triple of a few-shot task for relation r, *e.g.*, (h, r, t). Take the head entity h as a target, and we denote its one-hop neighbors as $\mathcal{N}_h = \{(r_{nbr}, e_{nbr}) | (h, r_{nbr}, e_{nbr}) \in \mathcal{G}'\}$. Here, $\mathcal{G}'$ is the background KG; r_{nbr}, e_{nbr} represent the neighboring relation and entity of h respectively. The aim of the proposed neighbor encoder is to obtain varied entity representations with $\mathcal{N}_h$ to exhibit their different roles when involved in different task relations. Figure 2(a) gives the details of the adaptive neighbor encoder, where `CeoOf` is the few-shot task relation and the other relations such as `MarryTo`, `ProxyFor` and `WorksWith` are the neighboring relations of the head entity `BillGates`.

As claimed in the introduction, the role of entity h can be varied with respect to the few-shot task relation r. However, few-shot task relations are always hard to obtain effective representations by existing embedding models that always require sufficient training data for the relations. Inspired by TransE (Bordes et al., 2013), we model the task relation embedding $\mathbf{r}$ as a translation between the entity embeddings $\mathbf{h}$ and $\mathbf{t}$, *i.e.*, we want $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ when the triple holds. The intuition here originates from linguistic regularities such as `Italy`$-$`Rome` $=$ `France`$-$`Paris`, and such analogy holds because of the certain relation `CapitalOf`. Under the translation assumption, we can obtain the embedding of few-shot task relation r given its entity pair (h, t):

$$\mathbf{r} = \mathbf{t} - \mathbf{h} \tag{1}$$

where $\mathbf{r}, \mathbf{t}, \mathbf{h} \in \mathbb{R}^d$; $\mathbf{t}$ and $\mathbf{h}$ are embeddings pre-trained on $\mathcal{G}'$ with current embedding model such as TransE; d denotes the pre-trained embedding dimension. Actually, the translation mechanism is not the only way to model the task relations. We leave the investigation of other KG embedding methods (Trouillon et al., 2016; Sun et al., 2019) to future work.

Intuitively, relations can reflect roles of an entity. As shown in Figure 1(a), the task relation `CeoOf` may be more related to `WorkWith` than `MarryTo`, since the first two exhibit a business role. That is to say, we can discern the roles of h according to the relevance between the task relation r and the neighboring relation r_{nbr}. Hence, we first define a metric function ψ to calculate their relevance score by a bilinear dot product:

$$\psi(r, r_{nbr}) = \mathbf{r}^\top \mathbf{W} \mathbf{r}_{nbr} + b \tag{2}$$

where $\mathbf{r}$ and $\mathbf{r}_{nbr}$ can be obtained by Eq. (1); both $\mathbf{W} \in \mathbb{R}^{d \times d}$ and $b \in \mathbb{R}$ are learnable parameters. Then, we obtain a role-aware neighbor embedding $\mathbf{c}_{nbr}$ for h by considering its diverse roles:

$$\mathbf{c}_{nbr} = \sum\nolimits_{e_{nbr} \in \mathcal{N}_h} \alpha_{nbr} \mathbf{e}_{nbr} \tag{3}$$

$$\alpha_{nbr} = \frac{\exp(\psi(r, r_{nbr}))}{\sum_{r_{nbr'} \in \mathcal{N}_h} \exp(\psi(r, r_{nbr'}))} \tag{4}$$

That means, when neighboring relations are more related to the task relation, $\psi(\cdot,\cdot)$ will be higher and the corresponding neighboring entities would play a more important role in neighbor embeddings.

In order to enhance entity embeddings, we simultaneously couple the pre-trained entity embedding $\mathbf{h}$ and its role-aware neighbor embedding $\mathbf{c}_{nbr}$. Then, h can be formulated as:

$$f(h) = \sigma(\mathbf{W}_1\mathbf{h} + \mathbf{W}_2\mathbf{c}_{nbr}), \tag{5}$$

where $\sigma(\cdot)$ denotes activation function, and we use Relu; $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d\times d}$ are learnable parameters. Entity representations obtained in this way shall 1) preserve individual properties made by the current embedding model, and 2) possess diverse roles adaptive to different tasks. The above procedure also holds for the candidate tail entity t.

4.2 Transformer Encoder for Entity Pairs

Based on enhanced entity embeddings, we are going to derive embeddings of entity pairs. Figure 2(b) gives the details of Transformer encoder for entity pairs. FAAN borrows ideas from recent techniques for learning dynamic KG embeddings (Wang et al., 2019). Given an entity pair in a task of r, *i.e*, $(h,t) \in \mathcal{D}_r$, we take each entity pair with its task relation as a sequence $X = (x_1, x_2, x_3)$, where the first/last element is head/tail entity, and the middle is the task relation. For each element x_i in X, we construct its input representation as:

$$\mathbf{z}_i^0 = \mathbf{x}_i^{\text{ele}} + \mathbf{x}_i^{\text{pos}} \tag{6}$$

where $\mathbf{x}_i^{\text{ele}}$ denotes the element embedding, and $\mathbf{x_i}^{\text{pos}}$ the position embedding. Both $\mathbf{x_1}^{\text{ele}}$ and $\mathbf{x_3}^{\text{ele}}$ are obtained from the adaptive neighbor encoder. We allow a position embedding for each position within length 3. After constructing all input representations, we feed them into a stack of L Transformer blocks (Vaswani et al., 2017) to encode X and obtain:

$$\mathbf{z}_i^l = \text{Transformer}(\mathbf{z}_i^{l-1}), l = 1, 2, \cdots, L. \tag{7}$$

where $\mathbf{z}_i^l$ is the hidden state of x_i after the l-th layer. Transformer adopts a multi-head self-attention mechanism, with each block allowing each element to attend to all elements with different weights in the sequence.

To perform the few-shot KG completion task, we restrict the mask solely to the task relation r (*i.e.* x_2), so as to obtain meaningful entity pair embeddings. The final hidden state $\mathbf{z}_2^L$ is taken as the desired representation for the entity pair in $\mathcal{D}_r$. Such representation encodes semantic roles of each entity, and thus helps discern fine-grained meanings of task relations associated with different entity pairs. For more details about Transformer, please refer to Vaswani et al. (2017).

4.3 Adaptive Matching Processor

To make predictions by comparing the query to references, we devise an adaptive matching processor considering different semantic meanings of the task relation. Figure 2(c) gives the details of adaptive matching processor.

In order to compare one query to K-shot references, we are going to obtain a general reference representation for the given reference set $\mathcal{S}_r$. Considering the various meanings of the task relation, we define a metric function $\delta(q_r, s_{rk})$ that measures the semantic similarity of the query q_r and the reference triple s_{rk}. For simplicity, we achieve $\delta(q_r, s_{rk})$ with simple but effective dot product:

$$\delta\left(q_r, s_{rk}\right) = \mathbf{q}_r \cdot \mathbf{s}_{rk} \tag{8}$$

Unlike current few-shot relational learning models that learn static representations when predicting different queries, we adopt attention mechanism to obtain a general reference representation $g\left(\mathcal{S}_r\right)$ adaptive to the query. This can be formulated as:

$$g\left(\mathcal{S}_r\right) = \sum\nolimits_{s_{rk}\in\mathcal{S}_r} \beta_k \mathbf{s}_{rk} \tag{9}$$

$$\beta_k = \frac{\exp(\delta\left(q_r, s_{rk}\right))}{\sum_{s_{rj}\in\mathcal{S}_r} \exp(\delta\left(q_r, s_{rj}\right))} \tag{10}$$

Here, β_k denotes the attention score of a reference; $s_{rk} \triangleq (h_k, t_k) \in \mathcal{S}_r$ denotes the k-th reference in the task of r, and $\mathbf{s}_{rk}$ is its embedding; $\mathbf{q}_r$ is the embedding of a query q_r in $\mathcal{Q}_r$. Both $\mathbf{s}_{rk}$ and $\mathbf{q}_r$ are obtained by Eq. (7), to capture their fine-grained meanings. Eq. (9) leads to the fact that references having similar meanings to the query would be more referential, making reference set $\mathcal{S}_r$ have an adaptive representation to different queries.

To make predictions, we define a metric function $\phi\left(q_r, \mathcal{S}_r\right)$ to measure the semantic similarity of the query q_r and the reference representation $\mathcal{S}_r$:

$$\phi\left(q_r, \mathcal{S}_r\right) = \mathbf{q}_r \cdot g\left(\mathcal{S}_r\right). \tag{11}$$

$\phi(\cdot)$ is expected to be large if the query holds, and small otherwise. Here, $\phi\left(\cdot,\cdot\right)$ can also be implemented with alternative metrics such as cosine similarity or Euclidean distance.

4.4 Model Training

With the adaptive neighbor encoder, the Transformer encoder and the adaptive matching processor, the overall model of FAAN is then trained on meta-training set $\mathcal{T}_{mtr}$. $\mathcal{T}_{mtr}$ is obtained by the following way. For each few-shot relation r, we randomly sample K-shot positive entity pairs from $\mathcal{T}$ as the reference set $\mathcal{S}_r$. The remaining entity pairs are utilized as positive query set $Q_r = \{(h_m, t_m)\}$. Then we construct a set of negative queries $Q_r^- = \{(h_m, t_m^-)\}$ by randomly corrupting the tail entity of (h_m, t_m), where $t_m^- \in \mathcal{E} \setminus \{t_m\}$. Then, the overall loss is formulated as:

$$\mathcal{L} = \sum_{r} \sum_{q_r \in Q_r} \sum_{q_r^- \in Q_r^-} \left[\gamma + \phi(q_r^-, \mathcal{S}_r) - \phi(q_r, \mathcal{S}_r)\right]_+ \tag{12}$$

where $[x]_+ = \max(0, x)$ is standard hinge loss, and γ is a margin separating positive and negative queries. To minimize $\mathcal{L}$, we take each relation in $\mathcal{T}_{mtr}$ as a task, and adopt a batch sampling based meta-training procedure proposed in (Zhang et al., 2020). To optimize model parameters in Θ and Transformer, we use Adam optimizer (Kingma and Ba, 2015), and further impose L_2 regularization on the parameters to avoid over-fitting.

5 Experiments

In this section, we conduct link prediction experiments to evaluate the performance of FAAN.

5.1 Datasets

We conduct experiments on two public benchmark datasets: NELL and Wiki[1]. In both datasets, relations that have less than 500 but more than 50 triples are selected to construct few-shot tasks. There are 67 and 183 tasks in NELL and Wiki, respectively. We use original 51/5/11 and 133/16/34 relations in NELL and Wiki, respectively, for training/validation/testing as defined in Section 3. Moreover, for each task relation, both datasets also provide candidate entities, which are constructed based on the entity type constraint (Xiong et al., 2018). More details are shown in Table 1.

Dateset	# Ent.	# Rel.	# Triples	# Tasks
NELL	68,545	358	181,109	67
Wiki	4,838,244	822	5,859,240	183

Table 1: Statistics of datasets. Each column represents the number of entities, relations, triples and tasks.

5.2 Comparision Methods

In order to evaluate the effectiveness of our method, we compare our method against the following two groups of baselines:

KG embedding method. This kind of method learns entity/relation embeddings by modeling relational structures in KG. We adopt five widely used methods as baselines: TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), SimplE (Kazemi and Poole, 2018) and RotatE (Sun et al., 2019). All KG embedding methods require sufficient training triples for each relation, and learn static representations of KG.

Few-shot relational learning method. This kind of method achieves state-of-the-art performance of few-shot KG completion on NELL and Wiki datasets. GMatching (Xiong et al., 2018) adopts a neighbor encoder and a matching network, but assumes that all neighbors contribute equally. FSRL (Zhang et al., 2020) encodes neighbors with a fixed attention mechanism, and applies a recurrent autoencoder to aggregate references. MetaR (Chen et al., 2019) makes predictions by transferring shared knowledge from the references to the queries based on a novel optimization strategy. All the above methods learn static representations of entities or references, ignoring their dynamic properties.

5.3 Implementation Details

We perform 5-shot KG completion task for all the methods. Our implementation for KG embedding baselines is based on OpenKE[2] (Han et al., 2018) with their best hyperparameters reported in the original literature. During training, all triples in background KG $\mathcal{G}'$ and training set, as well as few-shot reference triples of validation and testing set are used to train models. For few-shot relational learning baselines, we extend GMatching from original one-shot scenario to few-shot scenario by three settings: obtaining general reference representation by mean/max pooling (denoted as MeanP/MaxP) over references, or taking the reference that leads to the maximal similarity score to the query (denoted as Max). Because FSRL was reported in completely different experimental settings, we reimplement the

[1] https://github.com/xwhan/One-shot-Relational-Learning

[2] https://github.com/thunlp/OpenKE/tree/OpenKE-PyTorch

	NELL				Wiki			
	MRR	Hits@10	Hits@5	Hits@1	MRR	Hits@10	Hits@5	Hits@1
TransE (Bordes et al., 2013)	.174	.313	.231	.101	.133	.187	.157	.100
DistMult (Yang et al., 2015)	.200	.311	.251	.137	.071	.151	.099	.024
ComplEx (Trouillon et al., 2016)	.184	.297	.229	.118	.080	.181	.122	.032
SimplE (Kazemi and Poole, 2018)	.158	.285	.226	.097	.093	.180	.128	.043
RotatE (Sun et al., 2019)	.176	.329	.247	.101	.049	.090	.064	.026
GMatching (MaxP) (Xiong et al., 2018)	.176	.294	.233	.113	.263	.387	.337	.197
GMatching (MeanP) (Xiong et al., 2018)	.141	.272	.201	.080	.254	.374	.314	.193
GMatching (Max) (Xiong et al., 2018)	.147	.244	.197	.090	.245	.372	.295	.185
FSRL (Zhang et al., 2020)	.153	.319	.212	.073	.158	.287	.206	.097
MetaR (Chen et al., 2019)	.209	.355	.280	.141	.323	.418	.385	.270
FAAN (Ours)	**.279**	**.428**	**.364**	**.200**	**.341**	**.463**	**.395**	**.281**

Table 2: Results of 5-shot link prediction on NELL and Wiki. **Bold** numbers denote the best results of all methods.

model to make a fair comparison. We directly report the original results of MetaR with pre-trained embeddings to avoid re-implementation bias.

For all implemented few-shot learning methods, we initialize entity embeddings by TransE. The entity neighbors are randomly sampled and fixed before model training, and the maximum number of neighbors M is fixed to 50 on both datasets. The embedding dimensionality is set to 50 and 100 for NELL and Wiki, respectively. For FAAN, we further set the number of Transformer layers to 3 and 4, and the number of Transformer heads to 4 and 8, respectively. To avoid over-fitting, we also apply dropout to the neighbor encoder and the Transformer layer with the rate tuned in $\{0.1, 0.3\}$. The L_2 regularization coefficient is tuned in $\{0, 1e^{-4}\}$. The margin γ is fixed to 5.0. The optimal initial learning rate η for Adam optimizer is $5e^{-5}$ and $6e^{-5}$ for NELL and Wiki respectively, which is warmed up over the first 10k training steps, and then linearly decayed. We evaluate all methods for every 10k training steps, and select the best models leading to the highest MRR (described later) on the validation set within 300k steps. The optimal hyperparameters are tuned by grid search on the validation set.

5.4 Evaluation Metrics

To evaluate the performance of all methods, we measure the quality of the ranking of each test triple among all tail substitutions in the candidates: (h_m, r', t'_m), $t'_k \in \mathcal{C}_{h_m,r'}$. We report two standard evaluation metrics on both datasets: MRR and Hits@N. MRR is the mean reciprocal rank and Hits@N is the proportion of correct entities ranked in the top N, with $N = 1, 5, 10$.

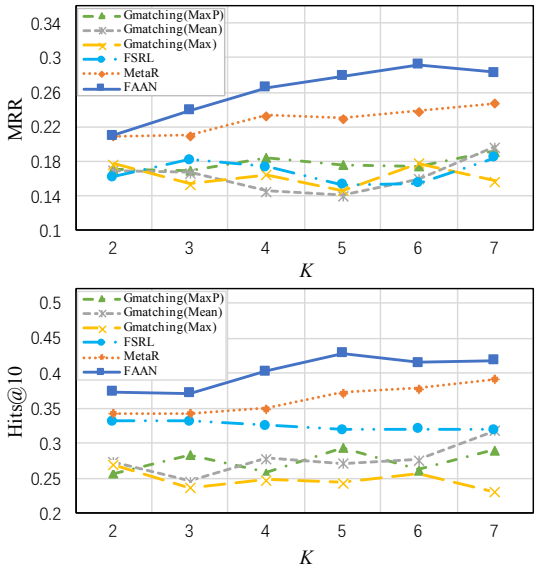

Figure 3: Impact of few-shot size K on NELL dataset.

5.5 Main Results in Link Prediction

The performance of all models on NELL and Wiki are shown in Table 2. The table reveals that:

(1) Compared to the traditional KG embedding methods, our model achieves better performance on both datasets. The experimental results indicate that our few-shot learning method is more suitable for solving few-shot issues.

(2) Compared to the few-shot learning baselines, our model also consistently outperforms them on both datasets in all metrics. Compared to the best performing baseline MetaR, FAAN achieves an improvement of 33.5%/20.6% in MRR/Hits@10 on NELL test data, and an improvement of 5.6%/10.8% on Wiki test data, respectively. It demonstrates that exploiting the dynamic properties of KG can indeed improve the

Variants	MRR	Hits@10	Hits@5	Hits@1
A1	.138	.295	.169	.072
A2	.209	.382	.294	.120
A3	.274	.411	.340	.199
B1	.235	.376	.301	.166
B2	.271	.413	.348	.195
C1	.219	.355	.287	.144
C2	.244	.395	.317	.171
C3	.212	.374	.295	.122
Ours	**.279**	**.428**	**.364**	**.200**

Table 3: Results of model variants on NELL dataset. **Bold** numbers denote the best results of all variants.

performance of few-shot KG completion.

5.6 Impact of Few-Shot Size

We conduct experiments to analyze the impact of few-shot size K. Figure 3 reports the performance of models on NELL data in different settings of K. The figure shows that:

(1) Our model outperforms all baselines by a large margin under different K, showing the effectiveness of our model in the few-shot scenario.

(2) An interesting observation is that a larger reference set does not always achieve better performance in the few-shot scenario. The reason is probably that few-shot scenario makes the performance sensitive to available references. Take the task relation `SubPartOf` in Figure 1(b) as an example. When making predictions for organization-related queries, injecting more location-related references is not necessarily useful. Even so, FAAN still gets relatively stable improvements compared to most baselines like GMatching and FSRL. The robustness to few-shot size comes from better reference embeddings generated by the adaptive aggregator.

5.7 Discussion for Model Variants

To inspect the effectiveness of the model components, we show results of experiments for model variants in Table 3:

(A) **Neighbor Encoder Variants:** In A1, we replace the encoder by mean pooling module used in GMatching. In A2, we aggregate neighbors with a fixed attention map as used in FSRL. In A3, we remove the embeddings of entities' own and encode them with only their neighbors. Experiments show that aggregating entity neighbors in an adaptive way and considering self-embedding can benefit the model performance.

(B) **Transformer Encoder Variants:** In B1, we replace the encoder by a concatenate operation on entity pairs as used in both GMatching and FSRL. In B2, we remove position embeddings in the Transformer encoder. Experiments indicate that the Transformer can effectively model few-shot relations, and position embeddings are also essential.

(C) **Matching Processor Variants:** In C1, we just obtain the embedding of reference set by averaging all reference representations. In C2, we only take the reference that is the most relevant to the query. In C3, we adopt the LSTM matching network as used in GMatching. Experiments indicate that our adaptive matching processor has superior capability in computing relevance between references and queries.

Tasks	Head Entity: Obama
HasSpouse	HasSpouse_Inv, HasFamilyMember, BornIn
Collaborate	PoliticianOffice, Graduated, ProxyOf
	Head Entity: Microsoft
ProxyFor	ProxyOf, Leader, AgentControls
CompeteWith	Acquired, Products, Collaborate

Table 4: The most contributive relation neighbors in different tasks. Top 3 relation neighbors are shown.

References	Query 1	Query 2
(Petersburg, Virginia)	.116	.230
(Vacaville, California)	.105	.306
(Prague, Czech)	.107	.314
(Cavaliers, NBA)	.208	.072
(L.A. Lakers, NBA)	.464	.078

Table 5: Attention weights of 5-shot references, given two queries: Query 1 `(C. Bulls, NBA)` and Query 2 `(Astana, Kazakhstan)`. The task relation of all entity pairs is `SubPartOf`. The references that are more related to the query achieve higher attention weights.

5.8 Case Study for Adaptive Attentions

To better understand the effects of adaptive attentions in the neighbor encoder and the matching processor, we conduct a case study. Table 4 provides the most contributive relation neighbors with the highest attention weights in different tasks. We can see that the contributive neighbors for each entity in both tasks are different. The entities tend to focus more on the neighbors that are related to the task. Table 5 shows attention weights of references given different queries. The attention map of references is varied for each query, and the queries focus more on the related references. We can see that the attention weights are higher for location-related ref-

RId	# Candidate	MRR		Hits@10	
		MetaR	FAAN	MetaR	FAAN
1	123	.971	**.974**	.971	**.986**
2	299	.371	**.533**	.453	**.766**
3	786	.211	**.352**	.524	**.610**
4	1084	.552	**.607**	.835	**.846**
5	2100	.522	**.595**	.643	**.735**
6	2160	.216	**.255**	.270	**.336**
7	2222	**.153**	.112	**.363**	.252
8	3174	.292	**.400**	.543	**.697**
9	5716	.066	**.084**	.133	**.168**
10	10569	**.054**	.050	.086	**.128**
11	11618	**.082**	.013	**.109**	.036

Table 6: Results of MetaR and FAAN for each relation (RId) in NELL testing data. # Candidate denotes the number of candidate entities. **Bold** numbers denote the best results of models.

erences when the query is location-related, while those are higher for organization-related references when the query is organization-related. This further indicates that our adaptive matching processor can aggregate references dynamically adaptive to the query, and benefits the matching process. All the above results further confirm our intuition described in the introduction.

5.9 Results on Different Relations

Besides the overall performance reported in the main results, we also conduct experiments to evaluate the performance of each task relation in NELL testing data. Table 6 reports the results of the best baseline model MetaR and our model FAAN. According to the table, we find that the results of both models on different task relations are of high variance. The reason may be that the number of candidate entities is different, and the relations with large candidate set are usually hard to make predictions. Even so, our model FAAN has better performance in most cases, which indicates that our model is robust for different task relations.

6 Conclusion

This paper proposes an adaptive attentional network for few-shot KG completion, termed as FAAN. Previous studies solve this problem by learning static representations of entities or references, ignoring their dynamic properties. FAAN proposes to encode entity pairs adaptively, and predict facts by adaptively matching references with queries. Experiments on two public datasets demonstrate that our model outperforms current state-of-art methods with different few-shot sizes. Our future work might consider other advanced methods to model few-shot relations, and exploiting more contextual information like textual description to enhance entity embeddings.

Acknowledgments

We would like to thank all the anonymous reviewers for their insightful and valuable suggestions, which help to improve the quality of this paper. This work is supported by the National Key Research and Development Program of China (No.2017YFB0803305) and the National Natural Science Foundation of China (No.61772151). This work is also supported by Beijing Advanced Innovation Center of Big Data and Brain Computing, Beihang University.

References

Kurt D. 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 ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*.

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, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4216–4225.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial*

Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pages 1811–1818.

Yan Duan, Marcin Andrychowicz, Bradly C. Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. 2017. One-shot imitation learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 1087–1098.

Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. Openke: An open toolkit for knowledge embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 139–144.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 687–696.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2020. A survey on knowledge graphs: Representation, acquisition and applications. *CoRR*, abs/2002.00388.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 4289–4300.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Zheng Li, Xin Li, Ying Wei, Lidong Bing, Yu Zhang, and Qiang Yang. 2019. Transferable end-to-end aspect-based sentiment analysis with selective adversarial learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4589–4599.

Hanxiao Liu, Yuexin Wu, and Yiming Yang. 2017. Analogical inference for multi-relational embeddings. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2168–2178.

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, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2395–2405.

Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3242–3250.

Anjing Luo, Pengpeng Zhao, Yanchi Liu, Jiajie Xu, Zhixu Li, Lei Zhao, Victor S. Sheng, and Zhiming Cui. 2019. Adaptive attention-aware gated recurrent unit for sequential recommendation. In *Database Systems for Advanced Applications - 24th International Conference, DASFAA 2019, Chiang Mai, Thailand, April 22-25, 2019, Proceedings, Part II*, volume 11447 of *Lecture Notes in Computer Science*, pages 317–332.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R. Voss, Heng Ji, Tarek F. Abdelzaher, and Jiawei Han. 2017. Cotype: Joint extraction of typed entities and relations with knowledge bases. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 1015–1024.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607.

Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3060–3067.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding

by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 1199–1208.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57:78–85.

Quan Wang, Pingping Huang, Haifeng Wang, Songtai Dai, Wenbin Jiang, Jing Liu, Yajuan Lyu, Yong Zhu, and Hua Wu. 2019. Coke: Contextualized knowledge graph embedding. *CoRR*, abs/1911.02168.

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

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, Brussels, Belgium, October 31 - November 4, 2018*, pages 1980–1990.

Bishan Yang and Tom M. Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1436–1446.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Shihui Yang, Jidong Tian, Honglun Zhang, Junchi Yan, Hao He, and Yaohui Jin. 2019. Transms: Knowledge graph embedding for complex relations by multidirectional semantics. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 1935–1942.

Chuxu Zhang, Huaxiu Yao, Chao Huang, Meng Jiang, Zhenhui Li, and Nitesh V. Chawla. 2020. Few-shot knowledge graph completion. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 3041–3048.