BERT-FKGC: Text-Enhanced Few-Shot Representation Learning for Knowledge Graphs

Jinlin Li[1,2], Zikang Wang[1,2,*], Linjing Li[1,2], Daniel Dajun Zeng[1,2]
[1]Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
{lijinlin2021, wangzikang2016, linjing.li, dajun.zeng}@ia.ac.cn

***Abstract*—In recent years, few-shot knowledge graph completion (FKGC) emerged as a prominent research problem, focused on utilizing a limited number of reference entity pairs to complete triples with unseen relations. Recent studies have attempted addressing this problem by modeling interactions between head and tail entities. However, existing FKGC methods represent semantics predominantly based on the neighborhood information of entities in the knowledge graph, thus can only infer the hidden and unobserved relations within the knowledge graph, limiting their reasoning capabilities. To overcome these limitations, we introduce text descriptions to FKGC and propose BERT-FKGC, a model capable of learning the integrated distribution of both the entity text descriptions and neighborhood information. By using a gating network that allows the model to dynamically select weights, our method can flexibly combine neighborhood information and textual descriptions. Besides addressing the prediction of unseen relations, our method is also capable of representing unseen entities. To validate the effectiveness of our model, we introduce a new dataset, FB15K-237-One, which includes textual descriptions for entities. We conduct extensive experiments on the FB15K-237-One dataset to validate the superiority of BERT-FKGC.**

***Index Terms*—few-shot learning, knowledge graph completion, pre-trained language model**

I. Introduction

Knowledge graphs are effective tools for managing and understanding massive amounts of data, such as YAGO [1], Wikidata [2] and Freebase [3]. In a knowledge graph, entities or concepts are represented as nodes, and their relations are depicted by edges connecting these nodes. Knowledge graphs have extensive applications in various fields, including LLMs [4], recommendation systems [5], and question-answering systems [6]. However, knowledge graphs often suffer from incompleteness, making knowledge graph completion a crucial research problem in the field. Representation learning is usually employed to address this issue. Nevertheless, these methods often require substantial amounts of training data, making them less effective in handling low-frequency relations. To tackle this challenge, few-shot knowledge graph completion (FKGC) [7] has emerged as a recent research hotspot.

* Corresponding author
This work was supported in part by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant #XDA27030100 and the National Natural Science Foundation of China under Grants #72293573, #72293575 and #62206282.

In contrast to traditional knowledge graph embedding(KGE) learning, FKGC aims to complete the incomplete triples with low-frequency relations. FKGC methods infer missing entities in the "query" triples by leveraging known knowledge, commonly referred to as the "support" knowledge or the "background" knowledge graph. The overall process can be summarized as utilizing the support information to ascertain the accuracy of the query. Few-shot learning in knowledge graphs has been proven to be an effective solution for predicting low-frequency relations and introducing new connections.

In recent years, there has been considerable interest in the development of few-shot learning methods for knowledge graphs. NP-FKGC [8] considers the inherent uncertainty of distributions, which proves to be crucial when learning from a limited set of reference triples. CIAN [9] places emphasis on the internal interactions among entities, facilitating the capture of their intricate relations. P-INT [10] directs its attention to the pathways of entities within the knowledge graph, facilitating the model's comprehension of the semantic associations between entities. However, existing methods are restricted by how they model background knowledge graphs, limiting their ability to infer only a limited number of relations implied in the original knowledge graph.

To tackle this issue, we propose a text-enhanced FKGC method, aiming to overcome this limitation by incorporating textual descriptions of entities. Specifically, we provide entities with two types of information: neighborhood information derived from the knowledge graph and textual descriptions of entities. Our model captures and combines these two types of information to generate the entity representation. On one hand, we employ graph convolutional networks to learn the distribution of neighborhood information for entities. On the other hand, we learn textual descriptions of entities using pre-trained language models and combine them with gate networks to form the final representation of entity pairs. We refer to this approach as the BERT-FKGC. Compared to previous methods, BERT-FKGC not only improves performance but also enables entities to conduct few-shot learning without relying on neighborhood information.

In general, our contributions can be summarized as the following aspects:

- We have created the FB15K-237-One dataset, which includes textual descriptions of entities, thereby providing a new dataset for FKGC.

- We have introduced a text-enhanced FKGC method BERT-FKGC, which combines pre-trained language models with FKGC, overcoming the limitations imposed by the background knowledge graph.
- We obtained the best results on the FB15K-237-One dataset, demonstrating the effectiveness of BERT-FKGC.

II. Related Work

A. Few-shot Knowledge Graph Completion

Existing knowledge graph few-shot learning methods can be categorized into three main groups: metric learning-based methods, meta learner-based methods, and dual-process theory-based methods.

Metric learning-based methods assess the accuracy of queries by evaluating the similarity of triple representations between the support set and the query set. Gmatching [7] is the pioneering method that introduced few-shot learning in knowledge graphs. It represents triples by aggregating neighborhood information. Afterwards, the similarity between the query set and the support set is calculated using LSTM, resulting in improved matching accuracy. FAAN [11] incorporates the Attention mechanism into the process of aggregating neighborhood information and represents the relation vector of triples using a Transformer network. The NP-FKGC [8] model integrates both the normalization flow and neural processes for modeling the intricate distribution of KG while estimating the degree of uncertainty. P-IN [10] utilizes the path from the head entity to the tail entity to express entity pairs information. CIAN [9] makes entity representation more distinguishable through the method of interaction between head and tail entities. InforMix-FKGC [12] encodes the various aspects of entity information and utilizes capsule networks to extract relation representation. InforMix-FKGC is the method most closely related to BERT-FKGC. However, it solely focuses on entity names and lacks a comprehensive analysis of the role of pre-trained language models in few-shot tasks.

MetaR [13] is a meta-learning-based model that optimizes relation representation by transferring relation-specific meta information between the query and support sets. GANA [14] further improves MetaR by adaptively utilizing the entity's neighborhood information and global-local framework.

Dual-process theory-based methods integrate both the summarization module and the reasoning module to simulate human cognitive processes, such as CogKR [15].

B. Knowledge Graph Completion with Language Models

Pre-trained language models have demonstrated remarkable performance across various natural language processing tasks and have garnered substantial attention in the field of knowledge graph completion (KGC) [16]. KG-BERT represents the pioneering effort in integrating KGC with pre-trained language models. It concatenates the textual descriptions of the head entity, relation, and tail entity to form the input for BERT. SimKGC [17] enhances performance through the incorporation of three different techniques for negative sample sampling. FTL-LM [18] improves the performance of KGC through path-based method for topology contexts learning and Expectation Maximization algorithm for logical rule distilling. StAR [19] utilizes a Siamese-style text encoder to process a triplet containing two contextual representations, and employs two parallel scoring strategies to learn contextualized and structured knowledge.

This paper examines the role of pre-trained language models in the context of FKGC and enhances their ability to infer knowledge using limited samples, while reducing the model's reliance on the background KG.

III. Background

A knowledge graph $\mathcal{G}$ consists of a series of triples $\{(h, r, t)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$. In our research, the set of relations $\mathcal{R}$ is divided into two subsets: $\mathcal{R}_{task}$ and $\mathcal{R}_b$, based on their respective frequencies. $\mathcal{R}_b$ consists of high-frequency relations, while $\mathcal{R}_{task}$ comprises low-frequency relations. We extract triples pertaining to $\mathcal{R}_{task}$ from the graph $\mathcal{G}$. The remaining set of triples will be used to construct a background knowledge graph denoted as $\mathcal{G}_b$. This study is centered on link prediction tasks, which involve predicting tail entities based on the head entity and an unseen relation: $(h, r, ?)$.

Definition 1 (Few-shot Knowledge Graph Completion). Given a unseen relation $r \in \mathcal{R}_{task}$ and its corresponding reference set $\mathcal{S}_r = \{(h_i^s, t_i^s)\}_{i=1}^K$, where $\mathcal{S}_r$ comprises a collection of correct triples associated with relation r. In addition, there exists a query set $\mathcal{Q}_r = \{(h^q, t^q/C_{h^q,r})\}$ about unseen relation r, where t^q is the correct tail entity and $C_{h^q,r}$ is the relevant candidate entities. According to the given size K of the reference set, we define few-shot tasks as k-shot tasks. Based on the reference set $\mathcal{S}_r$ and the query set $\mathcal{Q}_r$, we define the fundamental meta-relation tasks as $\mathcal{D}_r = \{\mathcal{S}_r, \mathcal{Q}_r\}$. The collection of all meta-relation tasks constitutes the meta-relation set, denoted as $\mathcal{T}_{mtr} = \{\mathcal{D}_r\}$. We partition all meta-relation tasks into three sets: training, validation, and testing. In order to avoid information leakage, the meta-relation tasks in the training, validation, and testing sets are fully disjointed.

During the training process, a specific number of meta-relation tasks are randomly chosen from the training set to train the model parameters θ. The objective of the training is to maximize the probability of the ground truth tail entity.

IV. Approach

This section introduces our method BERT-FKGC, comprising four components: (1) Graph encoder, which is an attention network that learns entity information in the background knowledge graph; (2) Text encoder for learning semantic descriptions of entities; (3) Relation encoder, which employs a gate network to regulate the importance of information from graph and text; (4) Matching processor, used to compare queries with provided references. The framework of BERT-FKGC is illustrated in Fig. 1.

A. Graph Encoder

Considering the complexity of the text encoder, we made efforts to reduce the complexity of the KG Encoder. The crux

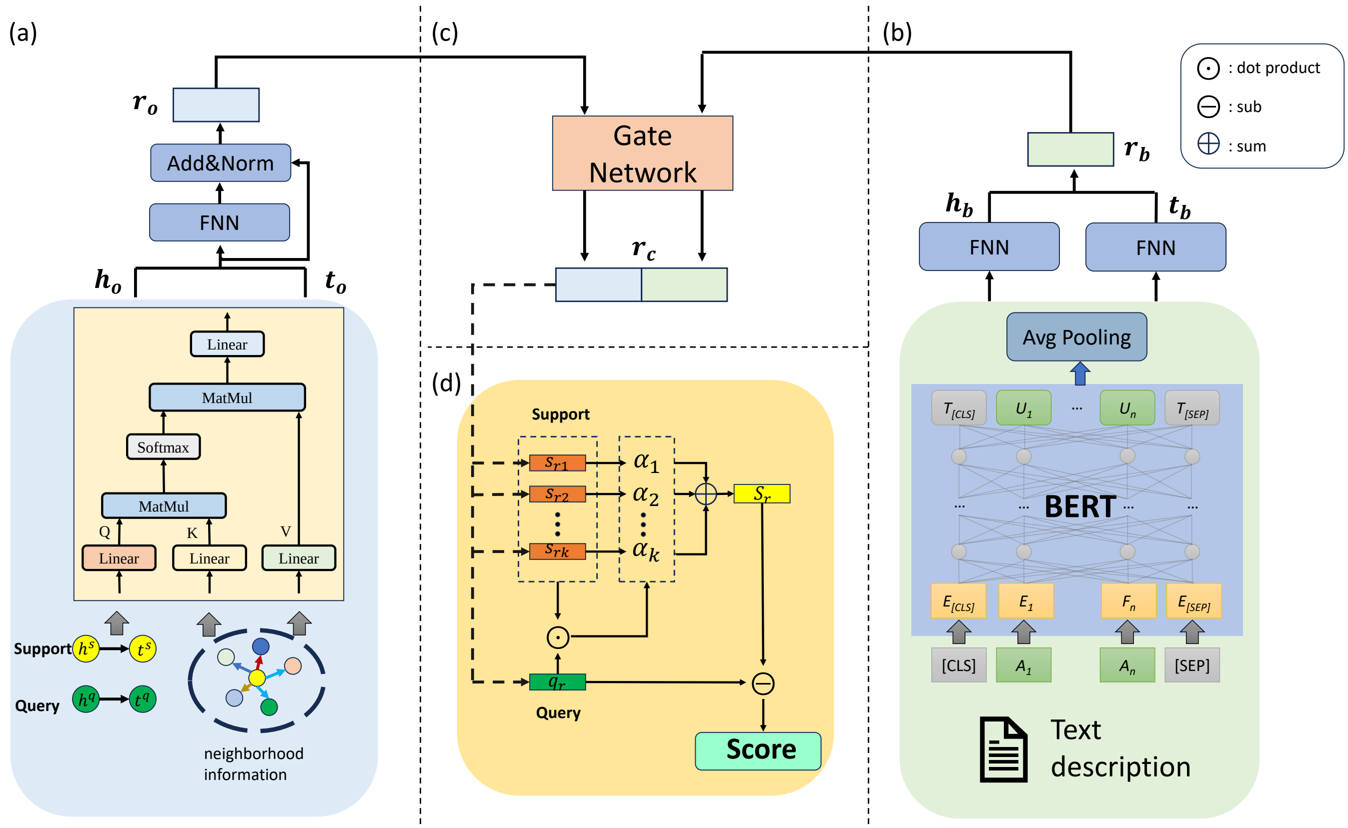

Fig. 1. The overview of BERT-FKGC: (a) Graph encoder for entity pairs; (b) Text encoder for entity descriptions; (c) Relation Encoder for merging head and tail entity representations; (d) Matching processor for similarity calculation. We employ superscripts to differentiate support facts from queries.

of previous studies involved employing graph neural networks to amalgamate the neighborhood information of entities within the background knowledge graph [20], and we adopted this approach.

We sampled a specific number of one-hop neighbors for entities from $\mathcal{G}_b$ and employed graph attention networks to dynamically learn entity representation. Specifically, for the task relation r, we obtain the associated (h, t) entity pairs. Consider the head entity h as an example, its one-hop neighborhood information is represented as $\mathcal{N}_h = \{(r_{hi}, e_{hi}) | (r_{hi}, e_{hi}) \in \mathcal{G}_b\}$. Based on the research of CIAN [9], we have ensured the interactivity between entity pairs while reducing network parameters. Although the approach of CIAN [9], which employs a two-layer attention network to represent entity vectors, overlooks the impact of reference triples on querying triples representation. Our aim is to employ a layer of attention network to accomplish adaptive entity extraction through task relations. The Q of the attention network consists of two parts: one part comes from the entity pairs in the query, and the other part comes from the entity pairs in the reference:

$$Q^q = W_Q\left([\boldsymbol{t^q} - \boldsymbol{h^q}; \boldsymbol{t^s} - \boldsymbol{h^s}]\right), \tag{1}$$

where $W_Q \in \mathbb{R}^{2d \times 2d}$ is a trainable parameter, d is the embedding dimension of entities and relations. $[\cdot\,;\,\cdot]$ denotes the concatenation operator. Superscripts are utilized when needed to differentiate between support and queries. For example, the query input feeds into the attention network denoted as Q^q. Equation (1) is motivated by the fact that the reference entity pairs encompass semantic information regarding the task relation r, while also accounting for the influence of querying entity pairs. The calculation method used to determine the difference is derived from the TransE [21], which enables the modeling of relation representations at a broader level. To maintain consistency in the process of modeling and querying reference triples, the definition of Q with respect to reference triples is established as follows:

$$Q^s = W_Q\left([\boldsymbol{t^s} - \boldsymbol{h^s}; \boldsymbol{t^s} - \boldsymbol{h^s}]\right). \tag{2}$$

Taking the head entity as an example, we obtain the other two parts of the input for input into the attention network from the entity's neighboring information, as follows:

$$K = W_K\left(W_N\left(\mathrm{ReLU}\left([r_h, e_h]\right)\right)\right), \tag{3}$$

$$V = W_V\left(W_N\left(\mathrm{ReLU}\left([r_h, e_h]\right)\right)\right), \tag{4}$$

where $W_K, W_V \in \mathbb{R}^{2d \times 5d}$ and $W_N \in \mathbb{R}^{5d \times 2d}$ are trainable parameters. The neighborhood information matrices r_h and e_h are formed by the embedding $\boldsymbol{r_{hi}}$ and $\boldsymbol{e_{hi}}$ from $\mathcal{N}_h$, where $r_h, e_h \in \mathbb{R}^{m \times d}$. Then, we use the method of dot-product attention to update the representation of h :

$$\boldsymbol{h_o} = W_o\left(\mathrm{softmax}\left(QK^T\right)V\right), \tag{5}$$

where $W_o \in \mathbb{R}^{d\times 2d}$ is a trainable parameter. We obtain t_o in a similar way. Then, we use $\boldsymbol{h_o}$ and $\boldsymbol{t_o}$ to represent the relation vector $\boldsymbol{r_o}$ between the head and tail entities.

$$\boldsymbol{r_p} = [\boldsymbol{h_o}, \boldsymbol{t_o}], \tag{6}$$

$$\boldsymbol{r_o} = \mathrm{LayerNorm}(\boldsymbol{r_p}) + [\boldsymbol{h}; \boldsymbol{t}] + W_p\left(\boldsymbol{r_p}\right), \tag{7}$$

where r_o represents the final representation of entity pairs in the background knowledge graph information, and LayerNorm is a commonly used normalization layer in neural networks. In order to highlight the significance of the initial entity embedding, we introduced residual connections to (7).

B. Text Encoder

In contrast to prior approaches, text descriptions are provided for all entities, and pre-trained language models are employed to extract semantic information [22]. Since the meta-task relation is unknown to the model, there is no textual description available for the relation. Given a meta-task relation r, we have corresponding (h, t) entity pairs and their respective textual descriptions (h_d, t_d).

The semantic vector of each entity is obtained by performing average pooling on the last hidden layer of BERT [22]. SimKGC [17] utilizes two distinct pre-training models to encode the head and tail entities separately.

We do not construct the form [cls, prompt, h_d, sep, t_d] [23]. Instead, we followed the approach of SimKGC [17] and separately input the h_d and t_d into the pre-trained language model, which greatly reduces the runtime of the pre-trained model. Two separate pre-trained language models are employed in SimKGC to individually encode the head and tail entities. Our approach simplified the SimKGC method through parameter sharing in the pre-training model and differentiation of entity encoding solely at the final fully connected layer. The semantic vector of each entity is derived by pooling the average of the last hidden layer of BERT. Regarding entity representation, the output of text encoder is as follows:

$$\boldsymbol{h_b} = \mathrm{ReLU}(W_b^1\left(\boldsymbol{h_{mean}}\right)), \tag{8}$$

$$\boldsymbol{t_b} = \mathrm{ReLU}(W_b^2\left(\boldsymbol{t_{mean}}\right)), \tag{9}$$

where W_b^1, W_b^2 are trainable parameters and $\boldsymbol{h_{mean}}, \boldsymbol{t_{mean}}$ are the output of the pre-training model. Then, we use $\boldsymbol{h_b}$ and $\boldsymbol{t_b}$ to obtain a text-based representation of the relation between the entity pair:

$$\boldsymbol{r_b} = W_b([\boldsymbol{h_b}; \boldsymbol{t_b}]), \tag{10}$$

where $W_b \in \mathbb{R}^{2d\times 2d}$ is a linear transformation matrix. To further reduce the complexity of the model, we considered two methods.The first approach involves substituting BERT with DistilBERT [24], effectively reducing the network's parameters. The second approach is to use an efficient fine-tuning method to control the number of trainable parameters in BERT. We have chosen LORA [25] for the fine-tuning process. The working principle of LoRA is to freeze the original model parameters while adding parallel network layers, focusing only on training the parameters of these newly added layers. We specifically implemented these two methods in our ablation experiments. For the regular experiment, we utilized Bert-base and froze the majority of parameters. During the inference process, we cache all entity vectors to further reduce the running time.

C. Relation Encoder

Through the graph encoder and the text encoder, we acquire relation representations for background knowledge graph modeling r_o and text description modeling r_b. As the features derive from structured graph information and unstructured text information, the approach is classified as a form of multimodal learning [26]. To integrate these distinct modalities of information, we designed a relation encoder network. We designed a gate network to dynamically consider the feature weights of the query triples:

$$\alpha_o = W_m(\boldsymbol{r_o^q} - \boldsymbol{r_o^s}), \tag{11}$$

$$\alpha_b = W_n(\boldsymbol{r_b^q} - \boldsymbol{r_b^s}), \tag{12}$$

where $W_m, W_n \in \mathbb{R}^{1\times 2d}$ are trainable parameters. The superscript of r is used to distinguish between references and queries. We have examined two modal fusion methods: feature fusion and decision fusion.

Feature fusion combines text features and graph neighborhood features to enhance the representation capability of the model. The first step connects the outputs of the graph encoder and the text encoder based on the weights of the gate network:

$$\boldsymbol{r_c} = [\alpha_o \boldsymbol{r_o}; \alpha_b \boldsymbol{r_b}]. \tag{13}$$

Next, a two-layer neural network is utilized to update the representation of r. Furthermore, normalization is performed to enhance the stability and reliability of the model.

Decision fusion involves inputting features from different modalities individually into an adaptive matching processor, which then calculates the matching scores. We introduce two trainable modal weight parameters, β_o and β_b, which allow the model to balance the strengths and weaknesses of different modal information. Furthermore, α_o and α_b are also involved in the calculation of the final decision weights.

D. Matching Processor and Loss Function

When there are multiple support triples, it becomes necessary to consider the impact of the different reference triples on the query. We can analyze the problem from two perspectives. Different support sets may have different semantic emphasis on task relations, so when the query is very similar to one of the references, it means the query is valid. Another perspective is that the relation vectors of the reference triples model are sampled from the real distribution of relations. We utilize the relation vectors of multiple triples to find the approximate distribution center and match it with the query. Inspired by FAAN [11] and the aforementioned perspectives, we introduce an attention mechanism to select reference entity pairs with greater informational value and perform denoising during the training process. Initially, compute the relevance between the query and various pairs of reference entities, followed by weighting the sum proportionally:

$$\alpha_i = \boldsymbol{q_r} \cdot \boldsymbol{s_{rk}}, \tag{14}$$

$$\boldsymbol{S_r} = \sum_{i=1}^{K} \mathrm{softmax}(\alpha_i) \boldsymbol{s_{rk}}, \tag{15}$$

where $\boldsymbol{s_{rk}}$ represents the relation of the k-th reference triple, and q_r represents the relation of the query. $\boldsymbol{S_r}$ is denoted as the comprehensive representation of all reference triples. Finally, we use Euclidean distance to measure the similarity score between the query triple representation and the reference representation:

$$\phi\left(\boldsymbol{q_r}, \boldsymbol{S_r}\right) = \|\boldsymbol{q_r} - \boldsymbol{S_r}\|^2 \tag{16}$$

We denote the score of negative samples as ϕ_- and the score of positive samples as ϕ_+. For each meta-relation r, a reference set is created by randomly sampling K pairs of positive entities from the scene. Additionally, a certain number of negative samples are generated for each positive sample. The model's loss function is defined as follows:

$$\mathcal{L} = \sum_r [\gamma + \frac{1}{m}\sum_{i=1}^{m}\phi_- - \frac{1}{n}\sum_{j=1}^{n}\phi_+], \tag{17}$$

where the margin γ is a hyperparameter to separate positive samples and negative samples.

V. Experiment

In this section, we will perform link prediction experiments to assess the effectiveness of BERT-FKGC.

A. Dataset and Metrics

Since our method relies on textual descriptions, we chose not to utilize the commonly employed Nell-one and wiki-one datasets as benchmarks [7]. Instead, we reconstructed the FB15K-237 [27] dataset to create FB15K-237-One. We identified triple relations that occurred fewer than 200 times as few-shot relations, and then divided them into 40/12/11 for training, validation, and testing. For the construction of the corresponding dataset for traditional KGE methods, we adopted Xiong's approach [7], which selects K triples for each few-shot relation and adds them to the training set.

To prevent information leakage issues, we removed all few-shot relations from the knowledge graph and ensured that the few-shot relations in the training/validation/testing sets were completely disjoint. It is important to highlight that previous datasets ensured the inclusion of all entities in the background knowledge graph. The core of existing FKGC methods is to evaluate the similarity of entity neighborhoods in the background knowledge graph, which requires each entity to have a certain number of adjacent nodes. To demonstrate our ability to overcome this constraint, not only are there unseen relations, but also unseen entities during the testing process, and we depend exclusively on textual descriptions to model semantic information.

To evaluate the performance of the models on FB15k-237-One, we used two metrics to measure the results: MRR (Mean Reciprocal Rank) and Hits@N. MRR represents the average reciprocal rank of the correct entities, while Hits@N indicates the proportion of correct entities ranked within the top N, where N is set to 1, 3, and 10.

B. Baselines

We compare BERT-FKGC with the following two groups of baselines:

KGE methods. We adopt four widely used methods as baselines: TransE [21], DistMult [28], ComplEx [29], and RotatE [30]. Due to the requirement of a large amount of training data for traditional KGE methods, their performance is limited in the case of few-shot relations. Therefore, the performance of KGE methods on this task is relatively poor.

Few-shot KGC We select six representative FKGC models as baselines, including GANA [14], FAAN [11], NP-FKGC [8], MetaR [13], CIAN [9], and GMatching [7]. These methods include both recent classic methods and state-of-the-art methods. We conducted comparative experiments using publicly available code provided by the authors, and adjusted hyperparameters for FB15K-237-One to achieve their optimal performance. To ensure fairness, we excluded P-INT [10] as a baseline model due to its significantly different criteria for determining performance. Due to InforMix-FKGC [12] not providing open source code, we did not use it as a baseline.

C. Implementation Details

We conducted experiments on FB15K-237-One using 1-shot, 3-shot, and 5-shot settings. We implemented the KGE method using the code and hyperparameters provided by RotatE [30]. The results were averaged over 5 rounds. In order to accelerate model training and not affect the prediction of high-frequency relations between entities in the background knowledge graph, we used the embeddings trained by TransE for entities and relations, and froze them during the training process. We set 100 as the embedded dimension. The maximum number of neighbors for an entity is set to 50, and the maximum length of the text description is set to 100. To avoid overfitting, the dropout rate in the network is set to $\{0.1, 0.2\}$ and the margin γ is set to 5.0. The initial learning rate is set to $1e^{-5}$, and the optimizer selects Adam. Setting the batch size to 64 and the negative sampling size to 10. We employ a linear preheating approach for the initial 3,000 steps, with a maximum training round of 20,000 steps. In metrics, we choose MRR as the model selection metric. There are a multitude of hyperparameters, and through empirical knowledge, we have assigned appropriate values to specific ones, including the number of neighbors and the dimension of embeddings.

D. Results

We conduct experiments on BERT-FKGC under 1-shot, 3-shot, and 5-shot conditions. The results are shown in Table 1. The results indicate that employing pre-trained language

TABLE I
LINK PREDICTION RESULTS ON FB15K-237-ONE.

		MRR			**Hits@10**			**Hits@3**			**Hits@1**		
		1-shot	3-shot	5-shot	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot
KGE	TransE	.245	.232	.236	.329	.334	.334	.285	.289	.290	.179	.150	.157
	DistMult	.191	.192	.193	.220	.222	.222	.186	.188	.191	.173	.172	.174
	ComplEX	.200	.196	.204	.238	.225	.246	.199	.193	.208	.176	.175	.178
	RotatE	.168	.161	.168	.242	.253	.251	.200	.197	.201	.114	.103	.112
FKGC	GMatching	.405	.412	.410	.442	.441	.468	.402	.411	.421	.382	.393	.378
	MetaR	.357	.371	.373	.382	.384	.388	.354	.371	.373	.334	.361	.364
	FAAN	.469	.532	.551	.495	.584	.633	.462	.531	.546	.438	.483	.513
	GANA	.328	.336	.357	.344	.356	.383	.323	.339	.349	.314	.328	.334
	NP-FKGC	.486	.561	.589	.512	.603	.644	.487	.558	.581	.449	.503	.544
	CIAN	.469	.532	.572	.497	.584	.625	.462	.531	.588	.447	.494	.531
	Ours	**.511**	**.569**	**.617**	**.583**	**.614**	**.663**	**.498**	**.599**	**.625**	**.465**	**.522**	**.583**

models to analyze text descriptions of entities are advantageous for completing knowledge graphs with limited samples. The results show that BERT-FKGC achieves the best performance on the FB15K-237-One dataset under different k-shot settings. This demonstrates the feasibility of utilizing both textual information and neighborhood information for few-shot knowledge graph completion. It can also be observed that the performance of MetaR and GANA on the FB15K-237-One dataset did not meet the expected standards. This could be attributed to the presence of unknown entities in the FB15K-237-One dataset, and these models heavily rely on information from the background graph. Recent approaches have enhanced model performance by improving the dataset and retraining embeddings. However, when using the same dataset, FAAN demonstrates comparable performance to the previous state-of-the-art models, while preserving low complexity.

Based on the number of head entities and tail entities, relations can be categorized into multiple types: 1-N, N-1, and N-N. For example, for a 1-N relation, a single head entity corresponds to multiple different tail entities. The difficulty of predicting relations varies greatly across different categories. We classify the meta-relation task of FB15K-237-One into these three categories to study the modeling capabilities of BERT-FKGC for complex relations. The results are shown in Table II. It can be seen that BERT-FKGC achieved the best results in all categories, which indicates that the introduction of text can model complex relations better. It is also evident that the prediction results for 1-N relations are generally poor, which may be due to two reasons: on the one hand, our training process focuses solely on extracting negative instances for the tail entity; on the other hand, the availability of reference samples is limited.

In order to showcase the breakthrough of BERT-FKGC in addressing the limitations of background knowledge graphs in model reasoning, we calculated the relevance metrics for the triples containing these unseen entities in the test set and the results are presented in Table III. According to the table, BERT-FKGC demonstrates superior performance compared to other methods. Certain methods lack the capability to

TABLE II
MRR RESULTS FOR COMPLEX RELATIONS.

Model	1-N	N-1	N-N
GMatching	.033	.641	.093
MetaR	.023	.589	.075
FAAN	.083	.840	.152
GANA	.012	.572	.058
NP-FKGC	.088	.893	.177
CIAN	.099	.864	.169
Ours	**.137**	**.914**	**.206**

complete link prediction tasks that involve unseen entities, consequently resulting in subpar overall prediction performance on the FB15K-237-One dataset. Given that unseen entities are predominantly found in 1-N complex relations, this further complicates the judgment process, leading to below-average performance on 1-N relations.

TABLE III
LINK PREDICTION RESULTS FOR TRIPLES INVOLVING UNSEEN ENTITIES

Model	MRR	Hits@10	Hits@3	Hits@1
GMatching	.011	.039	.012	.002
MetaR	.009	.007	.003	.001
FAAN	.285	.336	.302	.241
GANA	.008	.006	.002	.001
NP-FKGC	.289	.345	.311	.243
CIAN	.293	.318	.312	.246
Ours	**.348**	**.399**	**.353**	**.303**

E. Ablation study

Compared to existing models, BERT-FKGC improves performance by adding textual descriptions to entities. To investigate the impact of textual descriptions and neighborhood information on entity representation, we designed some variant models for BERT-FKGC. We conducted experiments from three perspectives:

TABLE IV
ABLATION STUDY OF LINK PREDICTION ON FB15K-237-ONE.

Model	MRR	Hits@10	Hits@3	Hits@1
Ours	**.617**	**.663**	**.625**	**.583**
only Graph	.575	.628	.599	.531
only Text	.589	.633	.601	.547

TABLE V
RESULTS OF DIFFERENT FUSION METHODS.

Model	MRR	Hits@10	Hits@3	Hits@1
Feature Fusion	.588	.647	.604	.529
Decision Fusion	**.617**	**.663**	**.625**	**.583**

(1) Impact of graph and text. We separately consider the impact of neighborhood information and textual descriptions on the performance of the model, and the results are shown in Table IV. The results exhibit a slight decrease when using only the graph encoder compared to the previous state-of-the-art model. In contrast, the influence of textual information on the model performance seems to be of greater significance. The proposed approach of information integration has also enhanced the model performance.

(2) Impact of feature fusion methods. We studied the impact of different feature fusion methods on the results. For specific settings, please refer to the previous section on the description of the gating network. The results are shown in Table V. The results of feature fusion are inferior to those of decision fusion. We attribute this to potential overfitting during the feature fusion process, which is also reflected in the changes in training loss. Conversely, few-shot learning necessitates a model with strong generalization ability. Further exploration of more effective fusion methods is left for future research.

(3) Impact of fine-tuning approaches. Due to GPU limitations, BERT-FKGC froze most of the network parameters of pre-trained language models in previous experiments. In this study, through the reduction of batch sizes and negative sample quantity, we performed full parameter fine-tuning on the model. Additionally, we also considered using DistilBERT and LoRA as variant models. The results from Table VI indicate that their performance is similar when using the same hyperparameter settings.

F. Visualization

We visualize the representations of relations to explore their distribution in the representation space. In this 5-shot experimental setup, we visualized the output representations of the relation encoder using principal component analysis. From Fig. 2, it can be observed that the representation of positive entity pairs learned by BERT-FKGC are closer to the vector distribution of the reference entity pairs, and they are significantly distinguishable from the negative samples.

TABLE VI
IMPACT OF DIFFERENT FINE-TUNING APPROACHES ON PRETRAINED MODELS.

Model	MRR	Hits@10	Hits@3	Hits@1
DistilBERT	.604	.656	**.611**	.564
BERT	.602	.655	.607	.561
LoRA	**.607**	**.659**	.610	**.569**

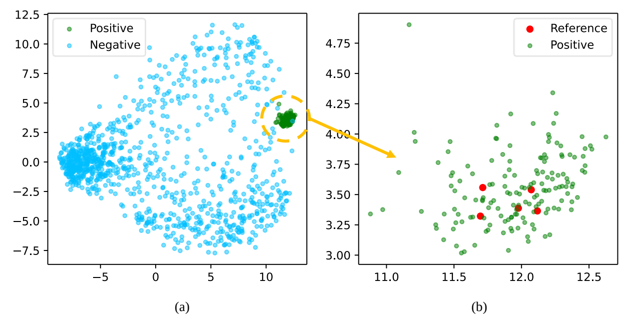

Fig. 2. The visualization of entity pair relation representations, where (b) represents a magnified view of the (a).

VI. CONCLUSION

This paper proposes a text-enhanced approach that combines pre-trained language models with traditional methods to achieve few-shot knowledge completion. First, we designed a single-layer attention network to adaptively aggregate neighborhood information of entities, considering both the mutual influence between head and tail entities and the impact of reference information on queries. Next, we employed pre-trained language models to extract textual information of entities and proposed some acceleration methods. Lastly, we used a gate network to control the weights of graph features and text features. Extensive experiments were conducted on the proposed FB15K-237-One dataset, and the results demonstrate that our method outperforms the state-of-the-art approaches. In the future, we will study more efficient fusion methods to enhance performance and explore acceleration methods for pre-trained language models in knowledge graph completion tasks.

ACKNOWLEDGMENT

We extend our sincere gratitude to the anonymous reviewers for their insightful and valuable suggestions, which have significantly enhanced the quality of this paper.

REFERENCES

[1] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 697–706.
[2] D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.
[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 1247–1250.

[4] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "Ernie: Enhanced language representation with informative entities," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1441–1451.
[5] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 950–958.
[6] X. Huang, J. Zhang, D. Li, and P. Li, "Knowledge graph embedding based question answering," in *Proceedings of the twelfth ACM international conference on web search and data mining*, 2019, pp. 105–113.
[7] W. Xiong, M. Yu, S. Chang, X. Guo, and W. Y. Wang, "One-shot relational learning for knowledge graphs," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 1980–1990.
[8] L. Luo, Y.-F. Li, G. Haffari, and S. Pan, "Normalizing flow-based neural process for few-shot knowledge graph completion," ser. SIGIR '23, 2023, p. 900–910.
[9] Y. Li, K. Yu, X. Huang, and Y. Zhang, "Learning inter-entity-interaction for few-shot knowledge graph completion," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 7691–7700.
[10] J. Xu, J. Zhang, X. Ke, Y. Dong, H. Chen, C. Li, and Y. Liu, "P-int: A path-based interaction model for few-shot knowledge graph completion," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 385–394.
[11] J. Sheng, S. Guo, Z. Chen, J. Yue, L. Wang, T. Liu, and H. Xu, "Adaptive attentional network for few-shot knowledge graph completion," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 1681–1691.
[12] Q. Li, J. Yao, X. Tang, H. Yu, S. Jiang, H. Yang, and H. Song, "Capsule neural tensor networks with multi-aspect information for few-shot knowledge graph completion," *Neural Networks*, vol. 164, pp. 323–334, 2023.
[13] M. Chen, W. Zhang, W. Zhang, Q. Chen, and H. Chen, "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, pp. 4217–4226.
[14] G. Niu, Y. Li, C. Tang, R. Geng, J. Dai, Q. Liu, H. Wang, J. Sun, F. Huang, and L. Si, "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*, 2021, pp. 213–222.
[15] Z. Du, C. Zhou, J. Yao, T. Tu, L. Cheng, H. Yang, J. Zhou, and J. Tang, "Cogkr: Cognitive graph for multi-hop knowledge reasoning," *IEEE Transactions on Knowledge & Data Engineering*, no. 01, pp. 1–1, aug 2019.
[16] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, "Unifying large language models and knowledge graphs: A roadmap," *IEEE Transactions on Knowledge and Data Engineering*, 2024.
[17] L. Wang, W. Zhao, Z. Wei, and J. Liu, "Simkgc: Simple contrastive knowledge graph completion with pre-trained language models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 4281–4294.
[18] Q. Lin, R. Mao, J. Liu, F. Xu, and E. Cambria, "Fusing topology contexts and logical rules in language models for knowledge graph completion," *Information Fusion*, vol. 90, pp. 253–264, 2023.
[19] B. Wang, T. Shen, G. Long, T. Zhou, Y. Wang, and Y. Chang, "Structure-augmented text representation learning for efficient knowledge graph completion," in *Proceedings of the Web Conference 2021*, 2021, pp. 1737–1748.
[20] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," *International Conference on Learning Representations*, 2018.
[21] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.
[22] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.
[23] L. Yao, C. Mao, and Y. Luo, "Kg-bert: Bert for knowledge graph completion," *arXiv preprint arXiv:1909.03193*, 2019.
[24] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *NeurIPS*, 2019.
[25] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2021.
[26] X. Chen, N. Zhang, L. Li, S. Deng, C. Tan, C. Xu, F. Huang, L. Si, and H. Chen, "Hybrid transformer with multi-level fusion for multimodal knowledge graph completion," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 904–915.
[27] K. Toutanova and D. Chen, "Observed versus latent features for knowledge base and text inference," in *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, 2015, pp. 57–66.
[28] B. Yang, S. W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, 2015.
[29] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *International conference on machine learning*. PMLR, 2016, pp. 2071–2080.
[30] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," in *International Conference on Learning Representations*, 2019.