PeerJ Computer Science

ShallowBKGC: a BERT-enhanced shallow neural network model for knowledge graph completion

Ningning Jia and Cuiyou Yao

School of Management and Engineering, Capital University of Economics and Business, Beijing, China

Submitted 9 January 2024
Accepted 23 April 2024
Published 15 May 2024

Corresponding author
Ningning Jia, jianingning@bupt.cn

Academic editor
José Alberto Benítez-Andrades

Additional Information and Declarations can be found on page 15

DOI **10.7717/peerj-cs.2058**

ABSTRACT

Knowledge graph completion aims to predict missing relations between entities in a knowledge graph. One of the effective ways for knowledge graph completion is knowledge graph embedding. However, existing embedding methods usually focus on developing deeper and more complex neural networks, or leveraging additional information, which inevitably increases computational complexity and is unfriendly to real-time applications. In this article, we propose an effective BERT-enhanced shallow neural network model for knowledge graph completion named ShallowBKGC. Specifically, given an entity pair, we first apply the pre-trained language model BERT to extract text features of head and tail entities. At the same time, we use the embedding layer to extract structure features of head and tail entities. Then the text and structure features are integrated into one entity-pair representation *via* average operation followed by a non-linear transformation. Finally, based on the entity-pair representation, we calculate probability of each relation through multi-label modeling to predict relations for the given entity pair. Experimental results on three benchmark datasets show that our model achieves a superior performance in comparison with baseline methods. The source code of this article can be obtained from https://github.com/Joni-gogogo/ShallowBKGC.

Subjects Artificial Intelligence, Computational Linguistics, Data Science, Natural Language and Speech, Neural Networks
Keywords BERT, Neural network, Knowledge graph, Knowledge graph completion

INTRODUCTION

Knowledge graphs (KGs) such as DBpedia (*Auer et al., 2007*), Freebase (*Bollacker et al., 2008*), NELL (*Carlson et al., 2010*), and Wikidata (*Vrandečič & Krötzsch, 2014*) are important resources for many artificial intelligence tasks including semantic search (*Feddoul, 2020*), recommendations (*Wu et al., 2022*) and question answering (*Li & Moens, 2022*). These KGs are composed of factual triplets, with each triplet (h, r, t) denotes the fact that relation r exists between head entity h and tail entity t. KGs can also be formalized as directed multi-relational graphs, where nodes correspond to entities and (labeled) edges represent the types of relationships among entities.

Although existing KGs usually contain more than billions of factual triplets, they still suffer from an incompleteness problem, *i.e.*, missing a large number of valid triplets (*Nguyen, 2020*). In particular, in English DBpedia 2014, 60% of person entities miss place-of-birth information, and 58% of the scientists have no facts about what they are known for

How to cite this article Jia N, Yao C. 2024. ShallowBKGC: a BERT-enhanced shallow neural network model for knowledge graph completion. **PeerJ Comput. Sci. 10:e2058 DOI 10.7717/peerj-cs.2058**

Krompaß, Baier & Tresp (2015). In Freebase, 71% of 3 million person entities miss place-of-birth information, 75% have no known nationality while 94% have no facts about their parents (*West et al., 2014*). Therefore, much efforts have focused on the knowledge graph completion (KGC) task, which aims to predict missing triplets in KGs by examining existing ones.

Knowledge graph embedding is a dominant approach for KGC, which maps entities and relations of a KG from a symbolic domain to a vector space and make predictions with their embeddings (*i.e.*, vectors). Among various knowledge graph embedding methods, deep neural network-based models such as convolutional neural network (*Dettmers et al., 2018*; *Dai Quoc Nguyen, Nguyen & Phung, 2018*), capsule network (*Nguyen et al., 2019*), graph neural network (*Schlichtkrull et al., 2018*; *Nguyen et al., 2022*; *Tong et al., 2023*), graph attention network (*Liang et al., 2023*; *Wang et al., 2023*) and generative adversarial network (*Li et al., 2023*) achieve state-of-the-art results.

In this article, we take a step back and propose a simple yet effective BERT-enhanced shallow neural network model for KGC, referred to as ShallowBKGC. Our motivation is based on the following considerations and observations: (1) deep neural network models are usually extended versions of simple shallow neural network models, and improving simple models can also produce corresponding improvements in complex deep models. (2) It has been demonstrated that neural networks with even one single hidden layer are universal approximators (*Demir, Moussallem & Ngomo, 2021*; *Ba & Caruana, 2014*), which means that shallow neural networks can learn almost any complex function previously learned by deep neural networks. Besides, the relatively low computational complexity of shallow neural networks makes them more suitable for large-scale KGs. Based on the above two observations, we prefer simple shallow neural network model instead of complex deep neural network model. It should be explained here that this work builds on a previous work (*Jia, 2022*), which we expand based on the next observation. *i.e.*, (3) Most neural network-based methods learn embeddings merely from structured triplets, ignoring rich text information contained in the entity name, which affects the accuracy of KGC. Recently, a pre-trained language model BERT (*Kenton & Toutanova, 2019*) has achieved great success on multiple natural language processing tasks. Some works (*Yao, Mao & Luo, 2019a*; *Kim et al., 2020*; *Zha, Chen & Yan, 2022*; *Wang et al., 2022*) represent entities and relations using their text information, and fine-tune BERT to infer the missing triplets. Although these works achieve appealing performance, they still fail to learn the structural information of a KG, and the fine-tune is computationally inefficient. In order to efficiently utilize both text and structural information, we apply BERT in the form of feature extraction to enhance the shallow neural network model to further improve the performance of KGC.

Our contributions in this article are summarized as follows:

- We propose ShallowBKGC, a BERT-enhanced shallow neural network model for KGC, which utilizes both text and structural information for this task.

- We introduce a pre-trained language model BERT in feature extraction manner to obtain text features of entities, thereby further improving the performance of KGC without retraining the proposed model.
- We conduct experiments on three benchmark datasets, and the experimental results demonstrate that our model achieves a superior performance in comparison with baseline methods.

RELATED WORK

Existing KGC methods can be roughly classified into four categories: translation-based models, tensor decomposition-based models, neural network-based models and pre-trained language/large language-based models.

Translation-based models

Translation-based models consider the relation between a head entity and a tail entity as a translation operation in the vector space and calculate the distance between the head entity vector and the tail entity vector to measure the plausibility of a triple. *Bordes et al. (2013)* present the initial translation-based model TransE, which learns low-dimensional and dense vectors for every entity and relation, so that relations correspond to translation vectors operating on vectors of entities. *Wang et al. (2014)* present TransH, which alleviates the complex relation problem in TransE by associating each relation with a relation-specific hyperplane. *Lin et al. (2015)* present a path-based TransE, named PTransE, which extends TransE by relation paths. *Nguyen et al. (2016)* present STransE that combines SE (*Bordes et al., 2011*) and TransE for KGC. *Sun et al. (2019)* present RotatE, which defines each relation as a rotation from head entity to tail entity in the complex space. *Le, Huynh & Le (2021)* present RotatH that combines RotatE and TransH for KGC.

Models of this category have the advantages of simplicity, intuitiveness, and high computational efficiency. However, research has shown that they have limitations in expressive power and are not suitable for non-Euclidean spaces.

Tensor decomposition-based models

Tensor decomposition-based models use triangular norm to measure the plausibility of triplets. *Yang et al. (2015)* present DistMult, which considers triplets as tensor decomposition and constrains all relation embeddings to be diagonal matrices. ComplEx (*Trouillon et al., 2016*) extends DistMult to the complex space to better model asymmetric and inverse relations. *Balažević, Allen & Hospedales (2019)* present TuckER, which performs KGC based on tucker decomposition of binary tensors of known triplets. Inspired by the tucker decomposition of order-4 tensors, *Shao et al. (2022)* present a tensor decomposition model for temporal KGC. *Zhang et al. (2024)* extend tensor decomposition methods to temporal KGC.

Models of this category are proficient in capturing complex relations between entities and relations in KGs. However, as the scale of KGs grows, the computational complexity of these models may escalate rapidly.

Neural network-based models

Various neural networks have been widely explored for KGC and achieved promising performance. *Dettmers et al. (2018)* present a multi-layer convolutional model ConvE, which explores convolutional neural network for KGC, and uses 2D convolution over embeddings to predict missing triplets in a KG. *Shang et al. (2019)* present an end-to-end graph structure-aware convolutional networks model SACN that combines graph convolutional network (GCN) and ConvE for KGC. *Dai Quoc Nguyen, Nguyen & Phung (2018)* present ConvKB, which utilizes convolutional neural network to capture the global relationships among dimensional entries of entity and relation embeddings. CapsE (*Nguyen et al., 2019*) combines ConvKB with capsule network for both KGC and search personalization tasks. *Schlichtkrull et al. (2018)* present relational graph convolutional networks and apply them to KGC. *Vashishth et al. (2020)* present CompGCN, which leverages a variety of composition operations from knowledge graph embedding techniques to jointly embed both entities and relation in a graph. SHALLOM (*Demir, Moussallem & Ngomo, 2021*) and the prior version of the model proposed in this article ASLEEP (*Jia, 2022*) apply shallow neural network for KGC and achieve good performance while maintaining high efficiency.

Models of this category have significant advantages in semantic feature learning, and our proposed model belongs to this category. The main difference between them and our model is that most of them usually rely on more deeper and complex neural networks while our model employs shallow neural network, which is not computationally demanding and friendly to real-time applications. Although there are several models that use shallow neural networks for KGC, these models only use structural information and ignore the rich information contained in text. Under the premise of keeping the model as simple as possible, we consider both text and structural information for KGC.

Pre-trained language/large language-based models

Pre-trained language models and large language models have received widespread attention in many natural language processing tasks, including KGC. *Yao, Mao & Luo (2019a)* explore the pre-trained language model BERT for KGC. StAR (*Wang et al., 2021*) extends KG-BERT by taking into account structural information for KGC. *Yao et al. (2023)* present KG-LLM, which investigates large language models, including ChatGLM (*Du et al., 2022*) and LLaMA (*Touvron et al., 2023*) for KGC. *Yang, Fang & Zhou (2023)* present a constrained-prompt KGC based on large language model. *Zhang et al. (2023)* present KoPA, which integrates pre-trained KG structural features with large language model for KGC.

Models of this category achieve great success in KGC. However, these models usually require diverse fine-tuning strategies, and mostly cost much time in training and inference. It should be pointed out that the model we proposed also uses the pre-trained model BERT. The difference from the existing model is that in order to keep the model as simple as possible, we use BERT in a feature extraction manner, that is, the parameters in BERT are not involved in training.

OUR PROPOSED MODEL

Problem formulation

A KG is a type of multi-relational directed graph that typically consists of a collection of triplets in the form of (h, r, t). It can be formally defined as $\mathscr{G} = (\mathscr{E}, \mathscr{R}, \mathscr{T})$, where $\mathscr{T}$ represents the set of all triplets, $\mathscr{E}$ and $\mathscr{R}$ represent the sets of all entities and relations respectively.

The objective of the KGC task is to predict missing relations in $\mathscr{G}$ based on the known triplets $\mathscr{T}$. In other words, the aim of KGC is to develop a model that accepts a query consisting of a head entity and a tail entity, $(h_i, ?, t_i)$, and ranks all candidate relations $r_c \in \mathscr{R}$ to resolve the query (*Lovelace & Rose, 2022*). An effective KGC model should enable correct candidates to have higher rankings than incorrect candidates.

Model overview

Our proposed model ShallowBKGC takes as input an entity pair, and outputs the probability that each relation exists between the two entities. As illustrated in Fig. 1, our model consists of three key steps: (1) entity feature extraction, (2) entity-pair representation, and (3) multi-label relation modeling. The detailed calculation process of each step is as follows.

Entity feature extraction

Given an entity pair (h, t), our model extracts the features of the head and tail entities by taking into account both text and structural information.

For text information, we apply the pre-trained language model BERT (*Kenton & Toutanova, 2019*), which has achieved great success in multiple natural language processing tasks, to extract text feature of the given entity. Figure 2 illustrates the framework of BERT for entity text feature extraction. Formally, given the text information of head and tail entities, *i.e.*, head entity name $h\_text = \{w_1^h, w_2^h, \ldots, w_N^h\}$ and tail entity name $t\_text = \{w_1^t, w_2^t, \ldots, w_N^t\}$ (since many entities lack descriptive information and introducing additional information will increase computational complexity, we only use the name information that each entity has as text information), we first add a special classification token (CLS) and a separate token (SEP) at the beginning and end of the entity name respectively to obtain the marked entity name. Then through the tokenizer we obtain the representation of the marked entity name. Finally, we put the representation into BERT to get the text features of the head and tail entities as follows:

$$\mathbf{C} = \mathrm{BERT}(\mathbf{CLS}, \mathbf{w}_1^{\mathrm{h}}, \mathbf{w}_2^{\mathrm{h}}, \ldots, \mathbf{w}_{\mathrm{N}}^{\mathrm{h}}, \mathbf{SEP}); \mathbf{h}_{\mathrm{t}} = \mathbf{C} \tag{1}$$

$$\mathbf{C} = \mathrm{BERT}(\mathbf{CLS}, \mathbf{w}_1^{\mathrm{t}}, \mathbf{w}_2^{\mathrm{t}}, \ldots, \mathbf{w}_{\mathrm{N}}^{\mathrm{t}}, \mathbf{SEP}); \mathbf{t}_{\mathrm{t}} = \mathbf{C} \tag{2}$$

where $\mathbf{C} \in \mathbb{R}^H$ is the hidden vector of the special token (CLS), which contains the features of the entire input text. Therefore, we use it as the text feature of the given entity.

For structural information, our model receives the IDs of the given head and tail entities, and extracts the structure features of them through embedding layer as follows:

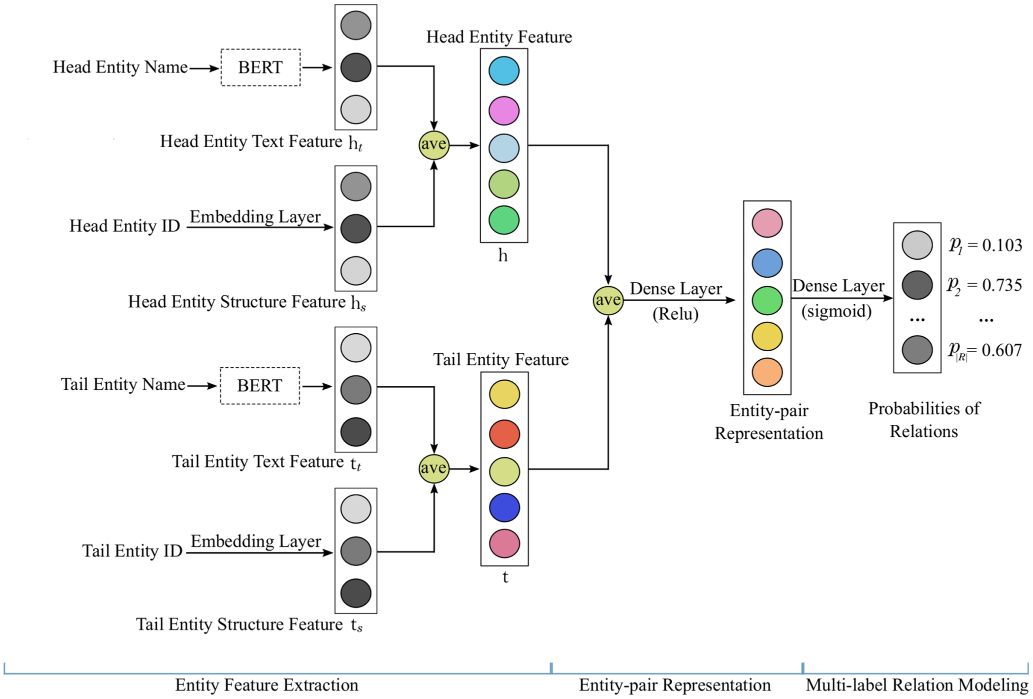

Figure 1 The architecture of our proposed model ShallowBKGC.
Full-size DOI: 10.7717/peerj-cs.2058/fig-1

$$\mathbf{h}_s = \text{Embedding\_layer}(\mathbf{h}_{\text{ID}}) \tag{3}$$

$$\mathbf{t}_s = \text{Embedding\_layer}(\mathbf{t}_{\text{ID}}) \tag{4}$$

where $\mathbf{h}_s \in \mathbb{R}^d$, $\mathbf{t}_s \in \mathbb{R}^d$ are embeddings of structural information corresponding to head and tail entities, respectively.

And then, we integrate entity text feature and entity structure feature through average operation, and get the entity feature as,

$$\mathbf{h} = \text{ave}(\mathbf{h}_\text{t}, \mathbf{h}_\text{s}) \tag{5}$$

$$\mathbf{t} = \text{ave}(\mathbf{t}_\text{t}, \mathbf{t}_\text{s}) \tag{6}$$

For the sake of computational convenience, and considering the consistency of tensor shapes, we intercept the first d columns of text features when fusing text and structure features of entities.

Entity-pair representation

After getting the features of entities, our model integrates head entity feature and tail entity feature into entity-pair representation through the average operation and a non-linear transformation as follows:

$$\mathbf{E} = \text{ReLU}(\mathbf{U} \cdot \text{ave}(\mathbf{h}, \mathbf{t})) \tag{7}$$

where $\mathbf{U} \in \mathbb{R}^{k \times k}$ is the transformation matrix, and ave() denotes the average function,

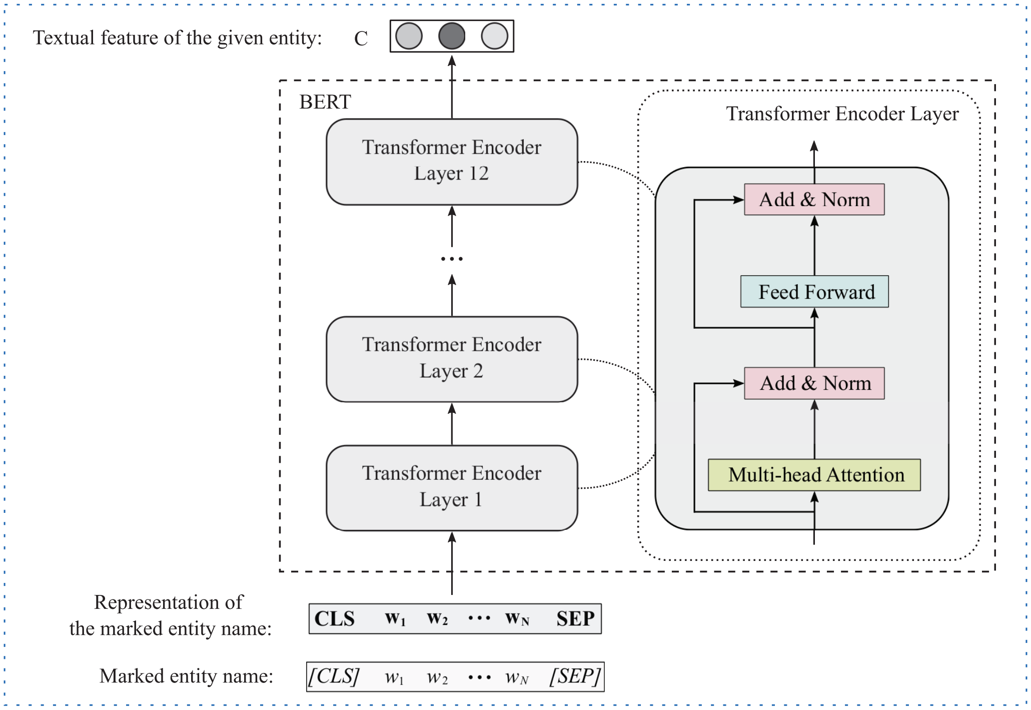
Textual feature of the given entity:
C
BERT
Transformer Encoder Layer 12
...
Transformer Encoder Layer 2
Transformer Encoder Layer 1
Transformer Encoder Layer
Add & Norm
Feed Forward
Add & Norm
Multi-head Attention
Representation of the marked entity name:
CLS W₁ W₂ ··· W_N SEP
Marked entity name:
[CLS] w₁ w₂ ··· w_N [SEP]

Figure 2 **The framework of BERT for entity text feature extraction.**
Full-size DOI: 10.7717/peerj-cs.2058/fig-2

which aims to obtain the comprehensive features among each dimensional of head and tail entity features.

Multi-label relation modeling

Since there may exist multiple relations between an entity pair, we model KGC as a multi-label learning problem. Based on the obtained entity-pair representation in the previous subsection, our model calculates the confidence scores for each relation as follows:

$$\mathbf{S} = \mathbf{V} \cdot \mathbf{E} \tag{8}$$

where $\mathbf{V} \in \mathbb{E}^{|R| \times k}$ is the collection of weight vectors for each relation. Afterwards, the sigmoid function is applied on each element of the score vector $\mathbf{S}$ to compute the probability of each relation to exist:

$$p_i = \frac{1}{1 + e^{-s_i}}, i = \{1, 2, \ldots, |R|\} \tag{9}$$

where $|R|$ denotes the number of relations.

Model training

We define the loss function using cross-entropy as follows:

$$L = -\sum_{i}^{|R|} y_i \mathrm{log}(\mathrm{p_i})) + (1 - \mathrm{y_i})\mathrm{log}(1 - \mathrm{p_i}) \tag{10}$$

where $y_i \in \{0, 1\}$ is the true value for relation i, p_i is the predicted probability value for relation i. The loss function is optimized with Adam (*Kingma & Ba, 2015*), and dropout (*Srivastava et al., 2014*) is employed for regularization.

EXPERIMENTS

Datasets

We evaluate our model ShallowBKGC on three benchmark datasets: WN18RR (*Dettmers et al., 2018*), FB15k-237 (*Toutanova & Chen, 2015*) and YAGO3-10 (*Mahdisoltani, Biega & Suchanek, 2015*). WN18RR and FB15k-237 are derived from the lexical KG WordNet (*Miller, 1995*) and the real-world KG Freebase (*Bollacker et al., 2008*), respectively. YAGO3-10 is a dataset containing general facts from Wikipedia. The experimental datasets statistics are shown in Table 1.

Evaluation metrics

We use mean rank (MR), mean reciprocal rank (MRR) and Hits@N as evaluation metrics, in which MR is the average rank of all test triplets, MRR is the average of the reciprocal ranks, and Hits@N is the percentage of test triplets that are ranked within top N. They are formally defined as follows:

$$\mathrm{MR} = \frac{1}{|Triplet_{test}|} \sum_{i=1}^{|Triplet_{test}|} rank_{triplet(i)} \tag{11}$$

$$\mathrm{MRR} = \frac{1}{|Triplet_{test}|} \sum_{i=1}^{|Triplet_{test}|} \frac{1}{rank_{triplet(i)}} \tag{12}$$

$$\mathrm{Hits@N} = \frac{|triplet(i) \in Triplet_{test} : rank_{triplet(i)} \leq N|}{|Triplet_{test}|} \tag{13}$$

where $|Triplet_{test}|$ is the number of test triplets, $triplet(i)$ is the i-th triplet.

Additionally, to evaluate the model efficiency, we measure the running time of the training phase. Record the average time of three epochs of the model on the dataset, in seconds. Our experimental platform is ModelArts, and the specific configuration selected is pytorch1.8-cuda10.2-cudnn7-ubuntu18.04, and a P100 GPU (16G).

Baseline methods

We compare our model against the following state-of-the-art KGC models, including translation-based models TransE and RotatE, tensor decomposition-based models DistMult and ComplEx, neural network-based models ConvE, SHALLOM and ASLEEP, and pre-trained language/large language-based models KG-BERT, KG-ChatGLM-6B, KG-LLaMA-7B and KG-LLaMA-13B. Below we briefly introduce these models.

- **TransE** (*Bordes et al., 2013*) is the initial translation-based model that views relations as translations from head entities to tail entities on the low-dimensional space.
- **DistMult** (*Yang et al., 2015*) is a typical tensor decomposition-based model that restricts n-by-n matrices representing relations to diagonal matrices.
- **ComplEx** (*Trouillon et al., 2016*) extends DistMult to the complex space.

Table 1 Statistics of datasets.

Dataset	#Entities	#Relations	#Train	#Validation	#Test
FB15k-237	14,541	237	272,115	17,535	20,466
WN18RR	40,943	11	86,835	3,034	3,134
YAGO3-10	123,182	37	1,079,040	5,000	5,000

Note:
#Entities denotes the number of all unique entities. #Relations denotes the number of all unique relations. #Train, #Validation and #Test denote the number of triplets contained in train set, validation set and test set, respectively.

- **RotatE** (*Sun et al., 2019*) is an efficient ranslation-based model that represents entities as complex vectors and relations as rotations.
- **ConvE** (*Dettmers et al., 2018*) is a deep neural network-based model that applies convolutional neural network for KGC.
- **SHALLOM** (*Demir, Moussallem & Ngomo, 2021*) is a shallow neural network-based model for KGC.
- **ASLEEP** (*Jia, 2022*) improves the way SHALLOM obtains entity pair representation, and is the prior version of our proposed model.
- **KG-BERT** (*Yao, Mao & Luo, 2019b*) is a pre-trained language-based model that firstly employs BERT to KGC.
- **KG-ChatGLM-6B, KG-LLaMA-7B, KG-LLaMA-13B** (*Yao et al., 2023*) are large language-based models that perform instruction tuning with ChatGLM (*Du et al., 2022*) and LLaMA (*Touvron et al., 2023*) for KGC.

Hyperparameter optimization

We select the hyperparameters of ShallowBKGC by grid search based on Hits@1 of the relation prediction task on the validation set of each dataset. We manually specify the hyperparameter ranges: embedding size among $\{50, 100, 150\}$, epochs among $\{50, 100, 150\}$, batch size among $\{512, 1000\}$, dropout rate among $\{0.25, 0.5\}$, and L_2-normalizer among $\{0.1, 0.01, 0.001\}$. Table 2 shows parameters values in the experiments.

Experimental results

In this section, we compare the performance of our model ShallowBKGC with that of the baseline methods on the widely used relation prediction task. The task of relation prediction is to complete a triplet (h, r, t) with r missing, *i.e.*, to predict the missing r given (h, t). From the relation prediction results shown in Tables 3–5, we summarize our key observations in the following section.

(1) The shallow neural network-based models, *i.e.*, our model ShallowBKGC and the baselines ASLEEP and SHALLOM, outperform the translation-based model TransE, the complex vector-based model ComplEx and RotatE, the deep neural network-based model ConvE, the pre-trained language-based model KG-BERT, and even the large language-based model KG-ChatGLM-6B, demonstrating the effectiveness of shallow neural network for KGC. For example, compared to TransE, DistMult, ComplEx and RotatE, our model

Table 2 **Hyperparameter values.**

Dataset	FB15k-237	WN18RR	YAGO3-10
Embedding size	50	100	50
Epochs	150	100	100
Batch size	1,000	512	512
L_2-normalizer	0.1	0.1	0.1

Table 3 **Relation prediction results on FB15k-237.**

Model	MR	MRR	Hits@1	Hits@3	RT
TransE (*Bordes et al., 2013*)*	1.352	0.966	0.946	0.984	31 s
DistMult (*Yang et al., 2015*)*	1.927	0.875	0.806	0.936	43 s
ComplEx (*Trouillon et al., 2016*)*	1.494	0.924	0.879	0.970	112 s
RotatE (*Sun et al., 2019*)*	1.315	0.970	0.951	0.980	>600 s
ConvE (*Dettmers et al., 2018*)*	–	0.667	0.562	0.732	33 s
SHALLOM (*Demir, Moussallem & Ngomo, 2021*)†	**1.106**	0.969	0.947	0.992	2 s
ASLEEP (*Jia, 2022*)	1.109	0.970	0.949	0.992	2 s
ShallowBKGC	1.108	**0.972**	**0.952**	**0.993**	3 s

Note:
The best score is in bold, while the second best score is underlined. Results marked * are taken from *Wang, Ren & Leskovec (2020)* and *Demir & Ngomo (2021)*, respectively. † denotes results from our re-implementation. RT is the abbreviation for running time.

Table 4 **Relation prediction results on WN18RR.**

Model	MR	MRR	Hits@1	Hits@3	RT
TransE (*Bordes et al., 2013*)*	2.079	0.784	0.669	0.870	36 s
DistMult (*Yang et al., 2015*)*	2.024	0.847	0.787	0.891	16 s
ComplEx (*Trouillon et al., 2016*)*	2.053	0.840	0.777	0.880	38 s
RotatE (*Sun et al., 2019*)*	2.284	0.799	0.735	0.823	>600 s
ConvE (*Dettmers et al., 2018*)*	–	0.353	0.143	0.405	35 s
SHALLOM (*Demir, Moussallem & Ngomo, 2021*)†	1.201	0.925	0.866	0.985	3 s
ASLEEP (*Jia, 2022*)	1.176	0.934	0.883	0.985	3 s
ShallowBKGC	**1.125**	**0.949**	**0.908**	**0.992**	4 s

Note:
The best score is in bold, while the second best score is underlined. Results marked * are taken from *Wang, Ren & Leskovec (2020)* and *Demir & Ngomo (2021)*, respectively. y denotes results from our re-implementation. RT is the abbreviation for running time.

ShallowBKGC achieves 45.8%, 44.4%, 45.2%, and 50.7% relative improvements in MR on WN18RR, respectively. Compared to ConvE, our model ShallowBKGC achieves 76.5% and 38.9% absolute improvements in Hits@1 on WN18RR and FB15k-237, respectively. Compared to KG-BERT and KG-ChatGLM-6B, our model ShallowBKGC achieves 0.2% and 11.7% absolute improvements in Hits@1 on YAGO3-10, respectively. It is worth mentioning that the result of KG-ChatGLM-6B is lower than that of KG-BERT. This

Table 5 Relation prediction results on YAGO3-10.

Model	MR	MRR	Hits@1	Hits@3	RT
KG-BERT (*Yao, Mao & Luo, 2019a*)*	–	–	0.681	–	>1,300 s
KG-ChatGLM-6B (*Yao et al., 2023*)*	–	–	0.566	–	–
KG-LLaMA-7B (*Yao et al., 2023*)*	–	–	**0.702**	–	–
KG-LLaMA-13B (*Yao et al., 2023*)*	–	–	0.696	–	–
SHALLOM (*Demir, Moussallem & Ngomo, 2021*)†	1.465	0.776	0.556	0.994	47 s
ASLEEP (*Jia, 2022*)†	1.388	0.813	0.630	0.996	47 s
ShallowBKGC	**1.301**	**0.837**	0.683	**0.996**	51 s

Note:

Results marked * are taken from *Yao et al. (2023)*. The dash (–) denotes values missing. The best score is in bold, while the second best score is underlined.

suggests that it is not the case that the more layers a model has or the newer the technology is, the better the results will be.

(2) Comparing our model ShallowBKGC with ASLEEP and SHALLOM, we can see that the MRR, Hits@1 and Hits@3 values of ShallowBKGC on the three datasets are better than ASLEEP and SHALLOM. This indicates that it is beneficial to take both text and structural information into account for KGC. Because the main difference between our model ShallowBKGC and the baselines ASLEEP and SHALLOM is that our model ShallowBKGC combines text and structural information for KGC, while ASLEEP and SHALLOM merely rely on structural information.

(3) The RTs of our model ShallowBKGC on three benchmark datasets significantly outperform the baselines, which shows the efficiency of our model. It should be noted that in order to minimize the impact of programming differences, we use OpenKE (*Han et al., 2018*) to reproduce the running times of TransE, DistMult, ComplEx and RotatE. The RTs of ConvE, SHALLOM, ASLEEP and KG-BERT are obtained from their corresponding source codes. Due to permission issues, the RTs of KG-ChatGLM-6B,KG-LLaMA-7B and KG-LLaMA-13B are missing. However, from the perspective of the number of layers and parameter scale, the RTs of these models are likely to be larger than KG-BERT. More formally, the time complexity of our model is $O(d_e)$ (where d_e represents the dimension of entities), which is the same as that of the baselines, except the language-based baselines. From *Wang et al. (2021)*, we can see that the most relevant language-based baseline KG-BERT's time complexity is $O(L_t^2|E|^2|R|)$, where L_t is the length of triple text, $|E|$ and $|R|$ are the numbers of entities and relations respectively. Additionally, the space complexity of our model is $O(L_e|E|d_{token} + |E|d_e)$, where L_e is the length of entity text, d_{token} is the dimension of entity text tokens.

(4) It is worth noting that the results of ConvE are significantly lower than those of the other models, probably because it relies on an improper pre-trained model for initialization, and is trained on entity prediction task (*i.e.*, given (h, r) predict t, or given (r, t) predict h) but tested on relation prediction task. It has been demonstrated that the initialization, hyperparameter optimization, and training strategies have significant effects on prediction performance (*Demir & Ngomo, 2021*; *Ruffinelli, Broscheit & Gemulla, 2020*).

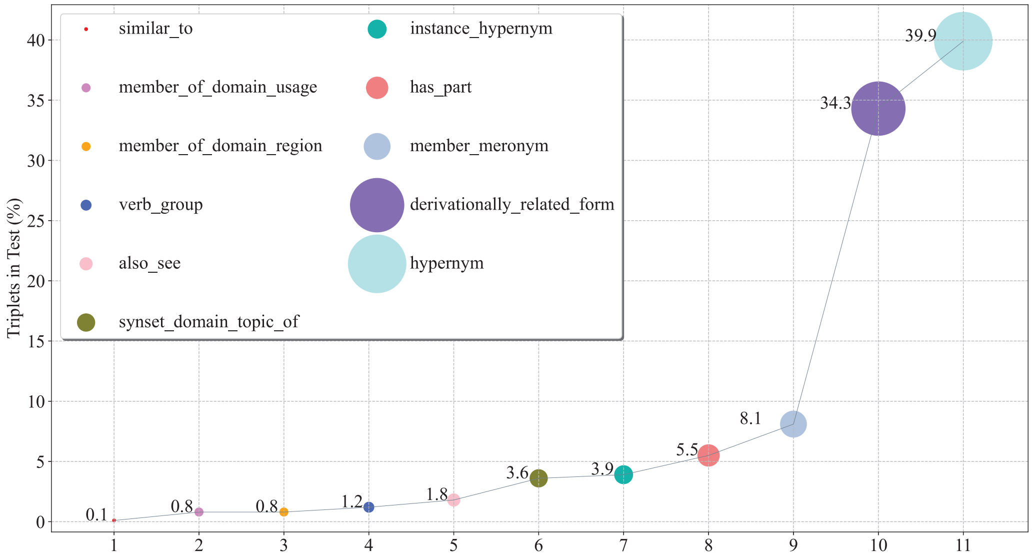

Figure 3 The percentage of triplets corresponding to relations on the WN18RR test set.
Full-size DOI: 10.7717/peerj-cs.2058/fig-3

In contrast, our model ShallowBKGC is as simple as possible, it does not require retraining the pre-trained model BERT, special hyperparameter optimization approach, or complex training strategy, thus minimizing model uncertainty.

Fine-grained performance analysis

To further verify the capacity of our model from a fine-grained perspective, we plot the percentage of each relation on WN18RR in Fig. 3, and report the Hits@N and MRR performance on each relation in Figs. 4 and 5, respectively.

From these figures, we can observe that:

(1) From the perspective of Hits@N, there are five relations, *i.e.*, *_similar_to*, *_member_of_domain_region*, *_instance_hypernym*, *_member_meronym*, and *_derivationally_related_form*, Hits@1 values exceed 90%. Ten relations Hits@3 values exceed 90%. In particular, there are four relations, *i.e.*, *_similar_to*, *_member_of_domain_region*, *_verb_group* and *_hypernym*, Hits@3 values reached 100%. Thus, the results are consistent with Table 4, which further demonstrates the effective of our model at a fine-grained level.

(2) There are eight relations with MRR values exceeding 90%. They are *_similar_to*, *_member_of_domain_usage*, *_member_of_domain_region*, *_synset_domain_topic_of*, *_instance_hypernym*, *_member_meronym*, *_derivationally_related_form* and *_hypernym*. Moreover, five of these eight relations Hits@1 values exceed 90%. This experimental result once again demonstrates the effectiveness of our model and the consistency of the results at a fine-grained level.

Ablation study

We conduct ablation studies to provide a more detailed analysis of the effectiveness of each part of our model. The models used for comparison are the following ones: (a)

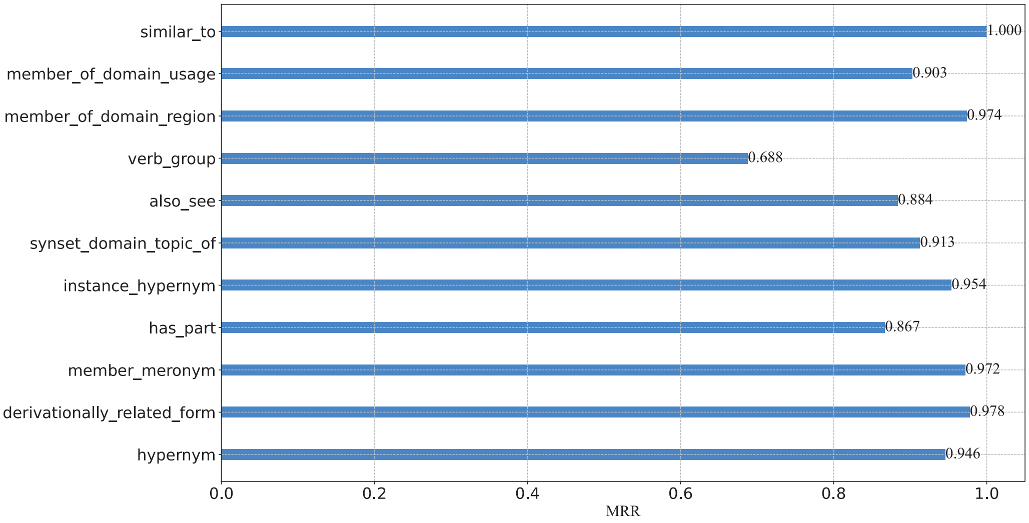

Figure 4 Hits@N results on the WN18RR test set. Full-size DOI: 10.7717/peerj-cs.2058/fig-4

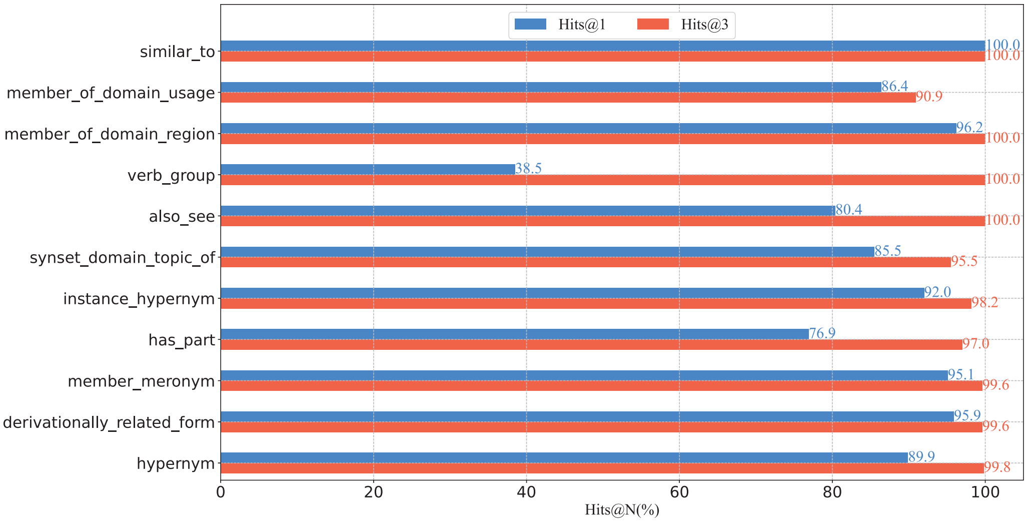

Figure 5 MRR results on the WN18RR test set. Full-size DOI: 10.7717/peerj-cs.2058/fig-5

ShallowBKGC-Text is the model that merely rely on text information, and (b) ShallowBKGC-Structure is the model that merely rely on structural information.

Table 6 shows the relation prediction results on three datasets, *i.e*., FB15k-237, WN18RR, and YAGO3-10. From which we can see that:

(1) ShallowBKGC outperforms ShallowBKGC-Text and ShallowBKGC-Structure in all three datasets, indicating that considering both text and structural information is beneficial to KGC. This is also consistent with the results of the previous experiments.

(2) ShallowBKGC-Structure achieves better results than ShallowBKGC-Text, the main reason is that we set the entity text feature obtained by BERT untrainable to reduce the computational complexity, this sacrifices the performance to a certain extent. It is worth further explaining that from Tables 5 and 6, we can see that ShallowBKGC-Text

Table 6 Ablation study tests on the three datasets.

Dataset	Model	MR	MRR	Hits@1	Hits@3	RT
FB15k-237	ShallowBKGC-Text	1.537	0.908	0.859	0.950	1 s
	ShallowBKGC-Structure	1.153	0.963	0.938	0.987	2 s
	ShallowBKGC	**1.108**	**0.972**	**0.952**	**0.993**	3 s
WN18RR	ShallowBKGC-Text	1.307	0.891	0.813	0.971	1 s
	ShallowBKGC-Structure	1.163	0.932	0.876	0.991	3 s
	ShallowBKGC	**1.125**	**0.949**	**0.908**	**0.992**	4 s
YAGO3-10	ShallowBKGC-Text	1.471	0.783	0.559	0.994	35 s
	ShallowBKGC-Structure	1.375	0.819	0.637	0.994	47 s
	ShallowBKGC	**1.301**	**0.837**	**0.683**	**0.996**	51 s

Note:
The best score of each dataset is in bold.

outperforms KG-ChatGLM-6B, considering that the latter has far more parameters than the former, but the experimental results are very close, which also reflects the efficiency of our model.

(3) ShallowBKGC-Text has the shortest RT because it has the fewest parameters. The RTs of ShallowBKGC-structure and ShallowBKGC are close, indicating that our model do not spend much time fusing text and structural information.

CONCLUSION

In this article, we propose a simple yet effective BERT-enhanced shallow neural network model for KGC that jointly considers text and structural information. Specifically, given an entity pair and the text information of the entities, our model first extracts the text features of the entities by BERT in a feature extraction manner, and extracts the structure features of the entities through the embedding layer. Then the text and structure features of the head and tail entities are integrated into an entity-pair representation through an average operation and a non-linear transformation, which aims to obtain the comprehensive and rich features of the entities. Finally, based on the entity-pair representation and considering that multiple relations may exist between entities, our model calculates the probability of each relation through multi-label modeling. Experimental results on three public datasets shown that our model achieves a superior performance in comparison with the baseline methods.

In the future, we plan to (1) further study the performance of our model on two KGC related tasks, *i.e*., triplet classification and entity prediction; (2) extend our model to temporal KGC and link prediction in social networks tasks; (3) explore the possibilities and performance of shallow neural networks on other tasks that can be organized into triplets.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their insightful comments and constructive suggestions.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This work was supported by the Capital University of Economics and Business under grant 01892254413027. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
Capital University of Economics and Business: 01892254413027.

Competing Interests

The authors declare that they have no competing interests.

Author Contributions

- Ningning Jia conceived and designed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, and approved the final draft.
- Cuiyou Yao performed the experiments, authored or reviewed drafts of the article, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The code is available at GitHub and Zenodo:

- https://github.com/Joni-gogogo/ShallowBKGC/tree/1.1.

- Joni-gogogo. (2024). Joni-gogogo/ShallowBKGC: knowledge graph completion (1.1). Zenodo. https://doi.org/10.5281/zenodo.10791673.

Supplemental Information

Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj-cs.2058#supplemental-information.

REFERENCES

Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z. 2007. DBpedia: a nucleus for a web of open data. In: *The Semantic Web*. Cham: Springer, 722–735.

Ba J, Caruana R. 2014. Do deep nets really need to be deep? *Advances in Neural Information Processing Systems* **27**:2654–2662.

Balažević I, Allen C, Hospedales T. 2019. TucKER: tensor factorization for knowledge graph completion. 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)*. 5185–5194.

Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. New York: ACM, 1247–1250.

Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O. 2013. Translating embeddings for modeling multi-relational data. *Advances in Neural Information Processing Systems* **26**:2787–2795.

Bordes A, Weston J, Collobert R, Bengio Y. 2011. Learning structured embeddings of knowledge bases. In: Burgard W, Roth D, eds. *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7–11, 2011*. Washington, D.C.: AAAI Press.

Carlson A, Betteridge J, Kisiel B, Settles B, Hruschka ER, Mitchell TM. 2010. Toward an architecture for never-ending language learning. In: *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Dai Quoc Nguyen TDN, Nguyen DQ, Phung D. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. In: *Proceedings of NAACL-HLT*. 327–333.

Demir C, Moussallem D, Ngomo A-CN. 2021. A shallow neural model for relation prediction. In: *IEEE 15th International Conference on Semantic Computing*. Piscataway: IEEE, 179–182.

Demir C, Ngomo AN. 2021. Out-of-vocabulary entities in link prediction. CoRR DOI 10.48550/arXiv.2105.12524.

Dettmers T, Minervini P, Stenetorp P, Riedel S. 2018. Convolutional 2D knowledge graph embeddings. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32, New York: ACM.

Du Z, Qian Y, Liu X, Ding M, Qiu J, Yang Z, Tang J. 2022. GLM: general language model pretraining with autoregressive blank infilling. In: Muresan S, Nakov P, Villavicencio A, eds. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22–27, 2022*. Stroudsburg: Association for Computational Linguistics, 320–335.

Feddoul L. 2020. Semantics-driven keyword search over knowledge graphs. In: Alani H, Simperl E, eds. *Proceedings of the Doctoral Consortium at ISWC 2020 co-located with 19th International Semantic Web Conference (ISWC 2020), Athens, Greece, November 3rd, 2020, volume 2798 of CEUR Workshop Proceedings*. Cham, Springer, 17–24.

Han X, Cao S, Xin L, Lin Y, Liu Z, Sun M, Li J. 2018. OpenKE: an open toolkit for knowledge embedding. In: *Proceedings of EMNLP*.

Jia N. 2022. ASLEEP: a shallow neural model for knowledge graph completion. In: Tanveer M, Agarwal S, Ozawa S, Ekbal A, Jatowt A, eds. *Neural Information Processing—29th International Conference, ICONIP 2022, Virtual Event, November 22-26, 2022, Proceedings, Part V, volume 1792 of Communications in Computer and Information Science*. Cham: Springer, 98–109.

Kenton JDM-WC, Toutanova LK. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of NAACL-HLT*. 4171–4186.

Kim B, Hong T, Ko Y, Seo J. 2020. Multi-task learning for knowledge graph completion with pre-trained language models. In: Scott D, Bel N, Zong C, eds. *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8–13, 2020*. New York: International Committee on Computational Linguistics, 1737–1743.

Kingma DP, Ba J. 2015. Adam: a method for stochastic optimization. In: Bengio Y, LeCun Y, eds. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*. San Diego: ICLR.

Krompaß D, Baier S, Tresp V. 2015. Type-constrained representation learning in knowledge graphs. In: *International Semantic Web Conference*. Cham: Springer, 640–655.

Le T, Huynh N, Le B. 2021. Link prediction on knowledge graph by rotation embedding on the hyperplane in the complex vector space. In: Farkas I, Masulli P, Otte S, Wermter S, eds. *Artificial Neural Networks and Machine Learning—ICANN 2021—30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part III, volume 12893 of Lecture Notes in Computer Science*. Cham: Springer, 164–175.

Li X, Ma J, Yu J, Zhao M, Yu M, Liu H, Ding W, Yu R. 2023. A structure-enhanced generative adversarial network for knowledge graph zero-shot relational learning. *Information Sciences* **629**:169–183 DOI 10.1016/j.ins.2023.01.113.

Li M, Moens M. 2022. Dynamic key-value memory enhanced multi-step graph reasoning for knowledge-based visual question answering. In: *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22–March 1, 2022*. Washington, DC: AAAI Press, 10983–10992.

Liang S, Zhu A, Zhang J, Shao J. 2023. Hyper-node relational graph attention network for multi-modal knowledge graph completion. *ACM Transactions on Multimedia Computing, Communications and Applications* **19(2)**:1–21 DOI 10.1145/3545573.

Lin Y, Liu Z, Luan H, Sun M, Rao S, Liu S. 2015. Modeling relation paths for representation learning of knowledge bases. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, 705–714.

Lovelace J, Rose C. 2022. A framework for adapting pre-trained language models to knowledge graph completion. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 5937–5955.

Mahdisoltani F, Biega J, Suchanek FM. 2015. YAGO3: a knowledge base from multilingual Wikipedias. In: *Seventh Biennial Conference on Innovative Data Systems Research, CIDR 2015, Asilomar, CA, USA, January 4–7, 2015, Online Proceedings*.

Miller GA. 1995. WordNet: a lexical database for English. *Communications of the ACM* **38(11)**: 39–41 DOI 10.1145/219717.219748.

Nguyen DQ. 2020. A survey of embedding models of entities and relationships for knowledge graph completion. In: *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*. 1–14.

Nguyen DQ, Sirts K, Qu L, Johnson M. 2016. Stranse: a novel embedding model of entities and relationships in knowledge bases. In: Knight K, Nenkova A, Rambow O, eds. *The Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12–17, 2016*. San Diego: The Association for Computational Linguistics, 460–466.

Nguyen DQ, Tong V, Phung D, Nguyen DQ. 2022. Node co-occurrence based graph neural networks for knowledge graph link prediction. In: *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. New York: ACM, 1589–1592.

Nguyen DQ, Vu T, Nguyen TD, Nguyen DQ, Phung D. 2019. A capsule network-based embedding model for knowledge graph completion and search personalization. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2180–2189.

Ruffinelli D, Broscheit S, Gemulla R. 2020. You CAN teach an old dog new tricks! On training knowledge graph embeddings. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020*.

Schlichtkrull M, Kipf TN, Bloem P, Berg RVD, Titov I, Welling M. 2018. Modeling relational data with graph convolutional networks. In: *European Semantic Web Conference*. Cham: Springer, 593–607.

Shang C, Tang Y, Huang J, Bi J, He X, Zhou B. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. *Proceedings of the AAAI Conference on Artificial Intelligence* **33(1)**:3060–3067 DOI 10.1609/aaai.v33i01.33013060.

Shao P, Zhang D, Yang G, Tao J, Che F, Liu T. 2022. Tucker decomposition-based temporal knowledge graph completion. *Knowledge-Based Systems* **238**:107841 DOI 10.1016/j.knosys.2021.107841.

Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* **15(1)**:1929–1958.

Sun Z, Deng Z-H, Nie J-Y, Tang J. 2019. Rotate: knowledge graph embedding by relational rotation in complex space. In: *International Conference on Learning Representations*.

Tong V, Nguyen DQ, Phung D, Nguyen DQ. 2023. Two-view graph neural networks for knowledge graph completion. In: *European Semantic Web Conference*. Cham: Springer, 262–278.

Toutanova K, Chen D. 2015. Observed versus latent features for knowledge base and text inference. In: *Proceedings of the 3rd Workshop on Continuous Vector Space Models and Their Compositionality*. 57–66.

Touvron H, Lavril T, Izacard G, Martinet X, Lachaux M-A, Lacroix T, Rozière B, Goyal N, Hambro E, Azhar F, Rodriguez A, Joulin Edouard Grave A, Lample G. 2023. LLaMA: open and efficient foundation language models. ArXiv preprint DOI 10.48550/arXiv.2302.13971.

Trouillon T, Welbl J, Riedel S, Gaussier É, Bouchard G. 2016. Complex embeddings for simple link prediction. In: *International Conference on Machine Learning*. PMLR, 2071–2080.

Vashishth S, Sanyal S, Nitin V, Talukdar P. 2020. Composition-based multi-relational graph convolutional networks. In: *International Conference on Learning Representations*.

Vrandečič D, Krötzsch M. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM* **57(10)**:78–85 DOI 10.1145/2629489.

Wang H, Ren H, Leskovec J. 2020. Entity context and relational paths for knowledge graph completion. CoRR DOI 10.48550/arXiv.2002.06757.

Wang B, Shen T, Long G, Zhou T, Wang Y, Chang Y. 2021. Structure-augmented text representation learning for efficient knowledge graph completion. In: Leskovec J, Grobelnik M, Najork M, Tang J, Zia L, eds. *WWW '21: The Web Conference 2021, Virtual Event/Ljubljana, Slovenia, April 19–23, 2021*. New York: ACM/IW3C2, 1737–1748.

Wang Y, Wang H, Lu W, Yan Y. 2023. HyGGE: hyperbolic graph attention network for reasoning over knowledge graphs. *Information Sciences* **630(1)**:190–205 DOI 10.1016/j.ins.2023.02.050.

Wang Z, Zhang J, Feng J, Chen Z. 2014. Knowledge graph embedding by translating on hyperplanes. In: Brodley CE, Stone P, eds. *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 –31, 2014, Québec City, Québec, Canada*. Washington, DC: AAAI Press, 1112–1119.

Wang L, Zhao W, Wei Z, Liu J. 2022. SimKGC: simple contrastive knowledge graph completion with pre-trained language models. In: Muresan S, Nakov P, Villavicencio A, eds. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22–27, 2022*. Dublin: Association for Computational Linguistics, 4281–4294.

West R, Gabrilovich E, Murphy K, Sun S, Gupta R, Lin D. 2014. Knowledge base completion via search-based question answering. In: *Proceedings of the 23rd International Conference on World Wide Web*. 515–526.

Wu B, Deng C, Guan B, Wang Y, Kangyang Y. 2022. Enhancing sequential recommendation via decoupled knowledge graphs. In: Groth P, Vidal M, Suchanek FM, Szekely PA, Kapanipathi P, Pesquita C, Skaf-Molli H, Tamper M, eds. *The Semantic Web—19th International Conference, ESWC 2022, Hersonissos, Crete, Greece, May 29–June 2, 2022, Proceedings, volume 13261 of Lecture Notes in Computer Science*. Cham: Springer, 3–20.

Yang R, Fang L, Zhou Y. 2023. CP-KGC: constrained-prompt knowledge graph completion with large language models. CoRR DOI 10.48550/arXiv.2310.08279.

Yang B, Yih SW-t, He X, Gao J, Deng L. 2015. Embedding entities and relations for learning and inference in knowledge bases. In: *Proceedings of the International Conference on Learning Representations*.

Yao L, Mao C, Luo Y. 2019a. KG-BERT: BERT for knowledge graph completion. ArXiv preprint DOI 10.48550/arXiv.1909.03193.

Yao L, Mao C, Luo Y. 2019b. KG-BERT: BERT for knowledge graph completion. CoRR DOI 10.48550/arXiv.1909.03193.

Yao L, Peng J, Mao C, Luo Y. 2023. Exploring large language models for knowledge graph completion. ArXiv preprint DOI 10.48550/arXiv.2308.13916.

Zha H, Chen Z, Yan X. 2022. Inductive relation prediction by BERT. In: *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22–March 1, 2022*. Washington, DC: AAAI Press.

Zhang F, Chen H, Shi Y, Cheng J, Lin J. 2024. Joint framework for tensor decomposition-based temporal knowledge graph completion. *Information Sciences* **654(2)**:119853 DOI 10.1016/j.ins.2023.119853.

Zhang Y, Chen Z, Zhang W, Chen H. 2023. Making large language models perform better in knowledge graph completion. CoRR DOI 10.48550/arXiv.2310.06671.