Knowledge and Information Systems (2024) 66:2683–2697
https://doi.org/10.1007/s10115-023-02040-z

REGULAR PAPER

# Simple knowledge graph completion model based on PU learning and prompt learning

Li Duan[1] · Jing Wang[1] · Bing Luo[1] · Qiao Sun[1,2]

Received: 27 April 2023 / Revised: 10 October 2023 / Accepted: 7 December 2023 /
Published online: 12 January 2024

## Abstract

Knowledge graphs (KGs) are important resources for many artificial intelligence tasks but usually suffer from incompleteness, which has prompted scholars to put forward the task of knowledge graph completion (KGC). Embedding-based methods, which use the structural information of the KG for inference completion, are mainstream for this task. But these methods cannot complete the inference for the entities that do not appear in the KG and are also constrained by the structural information. To address these issues, scholars have proposed text-based methods. This type of method improves the reasoning ability of the model by utilizing pre-trained language (PLMs) models to learn textual information from the knowledge graph data. However, the performance of text-based methods lags behind that of embedding-based methods. We identify that the key reason lies in the expensive negative sampling. Positive unlabeled (PU) learning is introduced to help collect negative samples with high confidence from a small number of samples, and prompt learning is introduced to produce good training results. The proposed PLM-based KGC model outperforms earlier text-based methods and rivals earlier embedding-based approaches on several benchmark datasets. By exploiting the structural information of KGs, the proposed model also has a satisfactory performance in inference speed.

**Keywords** Natural language processing · Knowledge graph completion · Prompt learning · Positive unlabeled learning

## 1 Introduction

Large-scale knowledge graphs (KGs), such as the Freebase [3], YAGO [25], and DBpedia [1], provide effective support for many important artificial intelligence tasks such as semantic search [12], recommendation [24, 37] and question answering [16]. With the rapid development of large language models, knowledge graph is still a powerful tool for knowledge

✉ Jing Wang
M21181005@nue.edu.cn

[1] College of Electronic Engineering, Naval University of Engineering, Wuhan 430033, China

[2] College of Computer Science and Technology, Zhejiang University, Hangzhou Zhejiang 310027, China

storage and knowledge processing, and it is still of great research significance. A KG is a multi-graph with entities as nodes and relations as edges. Each edge between two nodes in a KG is represented as a triple with a head entity, relation, and tail entity (h, r, t) indicating the relation between two entities, e.g., (Stanford, location, California). Although KGs are effective, most existing KGs suffer from data missing. This problem has inspired the knowledge graph completion (KGC) task, which aims at evaluating the rationality of potential triples in the KG and enriching the KG.

Many research efforts have been devoted to KGC. A common method called KG embedding maps each entity and relation to low-dimensional vectors and evaluates triples with those vectors [31]. Some typical models include the TransE [4], TransH [33], RotatE [26], and TuckER [2]. Text-based methods [29, 30, 36] use available textual information for KGC. Intuitively, text-based methods should outperform embedding-based methods because they introduce additional information. However, experiments on partial datasets show that text-based methods lag behind structure-based methods.

It is assumed that the key reason for this performance degradation is sampling. While negative sampling is important for KGC models, existing text-based methods only allow a limited amount of negative sampling due to the training cost of language models. For example, KEPLER [32] as a text-based method only trains for 30 epochs and utilizes a negative sample size of 1 due to the significant computational burden of RoBERTa. Conversely, embedding-based methods can collect more negative samples. RotatE model entails training for 1000 epochs on the Wikidata5M dataset with a negative sample size of 64. The RotatE model, as an embedding method, can be trained for 1000 periods on the Wikidata5M dataset with a negative sample size of 64. Both embedding-based methods and text-based methods will generate false-negative samples with a certain probability. However, the embedding-based method reduces the impact of false-negative samples on model training due to the large number of negative samples.

In this paper, inspired by the latest advances in positive and unlabeled (PU) learning and prompt learning, we propose a new knowledge graph completion model. PU learning is used to select the negative samples with the highest confidence in the candidate negative sample set to ensure the quality of negative samples in the case of limited negative sampling. Meanwhile, prompt learning is introduced to improve the reasoning ability of the model and to help judge negative examples. As shown in Fig. 1, we use a random replacement method to generate candidate negative samples for triples in positive samples. Prompt learning input is used to infer negative examples into the pre-trained language model. We then mix these negative examples with positive examples and use prompt learning to train the knowledge graph completion model. The entire process conforms to the two-step method in PU learning. To improve training results, the focal loss is introduced to adjust the effect of the ratio and difficulty of positive and negative samples to distinguish samples on the model. The trained model is then used to predict the rationality of triples. This method can perform effectively in multiple completion tasks. The contributions of this paper are summarized as follows.

1. We propose a KGC method based on PLMs. This method uses PU learning for the first time to solve the negative sampling problem of KGs.
2. The results of multiple benchmark datasets show that the proposed method can produce competitive results in link prediction tasks.
3. Compared to models with the same accuracy, the proposed model significantly reduces the inference time.

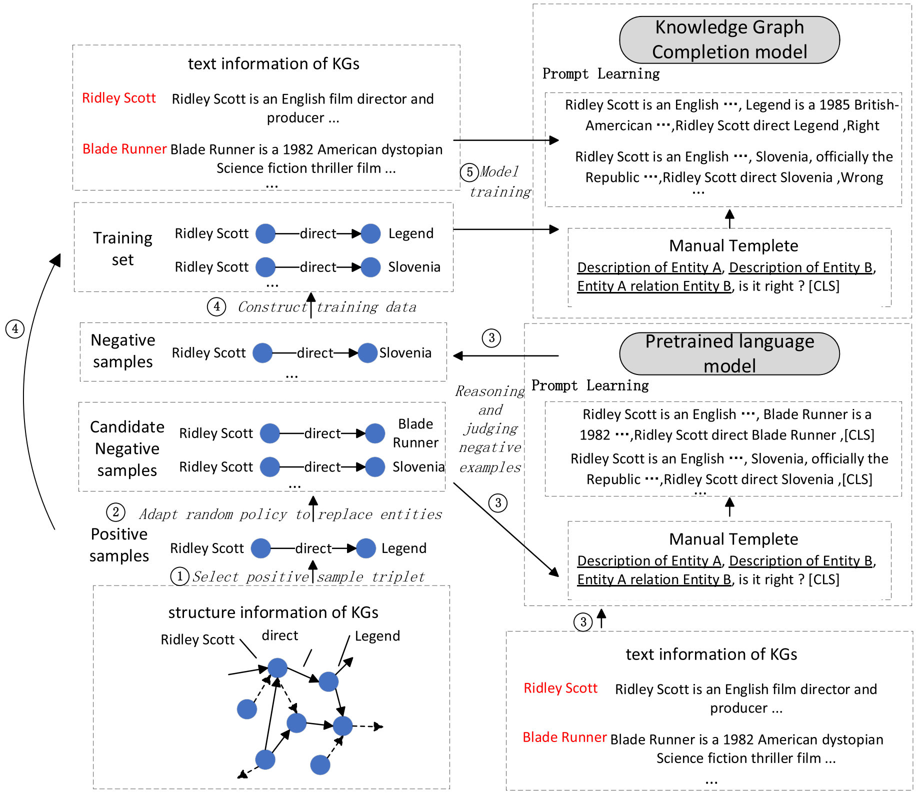
text information of KGs
Ridley Scott
Ridley Scott is an English film director and producer ...
Blade Runner
Blade Runner is a 1982 American dystopian Science fiction thriller film ...
...
Knowledge Graph Completion model
Prompt Learning
Ridley Scott is an English ···, Legend is a 1985 British-Amercican ···,Ridley Scott direct Legend ,Right
Ridley Scott is an English ···, Slovenia, officially the Republic ···,Ridley Scott direct Slovenia ,Wrong
...
5 Model training
Training set
Ridley Scott — direct → Legend
Ridley Scott — direct → Slovenia
...
Manual Templete
Description of Entity A, Description of Entity B, Entity A relation Entity B, is it right ? [CLS]
4 Construct training data
4
Negative samples
Ridley Scott — direct → Slovenia
...
3
Pretrained language model
Prompt Learning
Reasoning and judging negative examples
Candidate Negative samples
Ridley Scott — direct → Blade Runner
Ridley Scott — direct → Slovenia
...
Ridley Scott is an English ···, Blade Runner is a 1982 ···,Ridley Scott direct Blade Runner ,[CLS]
Ridley Scott is an English ···, Slovenia, officially the Republic ···,Ridley Scott direct Slovenia ,[CLS]
...
3
2 Adapt random policy to replace entities
Positive samples
Ridley Scott — direct → Legend
1 Select positive sample triplet
Manual Templete
Description of Entity A, Description of Entity B, Entity A relation Entity B, is it right ? [CLS]
3
structure information of KGs
Ridley Scott
direct
Legend
text information of KGs
Ridley Scott
Ridley Scott is an English film director and producer ...
Blade Runner
Blade Runner is a 1982 American dystopian Science fiction thriller film ...
...

**Fig. 1** Schematic diagram of negative example construction and model training

## 2 Related works

*KGC* helps complete the missing data in the existing KG by modeling the multi-relational data. Conventional methods like TransE [4] and TransH [33] use the structural information of the KG to accomplish inference. These methods regard the triple (h, r, t) as a specific relational transformation from the head entity h to the tail entity t. ComplEx [28] introduces multiple embedding to increase the expressive power of the model. RotatE [26] simulates the triple relational rotation in complex spaces. By using two vectors to represent each relation and adaptively adjusting the edge parameters in the loss function, scholars have implemented the coding of complex relational patterns [6]. Recently, scholars have tried to use additional textual information to improve KGC. The DKRL [34] uses convolutional neural networks (CNN) to encode the text, while the KG-BERT [36], StAR [29], BLP [9], and simKGC [30] all use PLMs to compute entity embedding. The performance of these methods has been improved but is still inferior to that of embedding-based methods on some datasets.

*PLMs* can be divided into two categories: feature-based methods and fine-tuning methods. Conventional word embedding methods such as the Word2Vec [21] and Glove [22] aim at taking a feature-based method to learn context-independent word vectors. In contrast to feature-based methods, fine-tuning methods such as the GPT [23] and BERT [11] first train a PLM on a large corpus of unlabeled text with language modeling objectives and then fine-tune the models for downstream tasks, which has triggered a paradigm shift in natural language processing.

*Prompt learning*, which directly models text probabilities, differs from conventional supervised learning. The training model takes an input x and predicts the output y. The template is used to modify the initial input x to a text string prompt with some padding slots so that these models can perform prediction tasks. The information is then padded to produce the final string Pt which is fed into the PLM to produce the output y. This framework is powerful and attractive for several reasons. It allows the language model to be pre-trained on massive amounts of raw text. By defining a new prompt function, the model can perform low-shot or even zero-shot learning, adapting to new scenarios with little or no label data. Prompt learning has achieved excellent results in natural language processing tasks such as text classification [13, 14], relation extraction [7, 15], named entity recognition [8], and question answering [18].

*PU learning* is based on scenarios where scholars have access only to positive examples and unlabeled data. PU learning has gained more attention because it appears naturally in applications such as medical diagnostics and KGC. Similar to general binary classification, the goal of PU learning is to train a classifier that enables classification according to target attributes. Most methods can be divided into three categories: the two-step methods [17], biased learning methods [20], and merged priority methods. Two-step methods, as the name suggests, involve two distinct steps. In the first step, reliable negative examples are identified, and in the second step, the classifier is trained based on the labeled positives and reliable negatives. The biased learning methods view unlabeled data as negative samples with label-like noise. The merged priority methods include postprocessing, preprocessing, and revising. Their main idea is to introduce class priority to modify traditional learning methods.

## 3 Methods

### 3.1 Symbols

The current study focuses on a knowledge graph (KG) that is a directed graph comprising entity E as the set of vertices and each edge represented by a triple (h, r, t), where h, r, and t denote the head entity, relation, and tail entity, respectively. The task of link prediction in KGs involves predicting the missing triples when the KG is incomplete. The widely adopted entity ranking evaluation protocol requires the sorting of all entities given h and r for tail entity prediction (h, r, ?), and the sorting of all entities given r and t for head entity prediction (?, r, t). In this work, an inverse triple (t, $r^{-1}$, h) is added for each triple (h, r, t) with $r^{-1}$ being the inverse relation of r. As a result, only tail entity prediction is required to deal with in this paper.

### 3.2 Model architecture

We propose a new PLM-based KGC model (PUPKGC), which can use the implicit knowledge as well as additional text description information in the PLM and infer new knowledge from structural information in KGs. As shown in Fig. 2, given a triple and its entity description, they are converted into a head entity prompt $\mathrm{PT}_{\mathrm{head}}$ and tail entity prompt $\mathrm{PT}_{\mathrm{tail}}$, a judgment sentence PJ, and an auxiliary prompt PA to input into the PLM. Formally, the final input text T of the PLM can be defined as T=$\mathrm{PT}_{\mathrm{head}}$ $\mathrm{PT}_{\mathrm{tail}}$ PJ PA [CLS]. The present study utilizes the

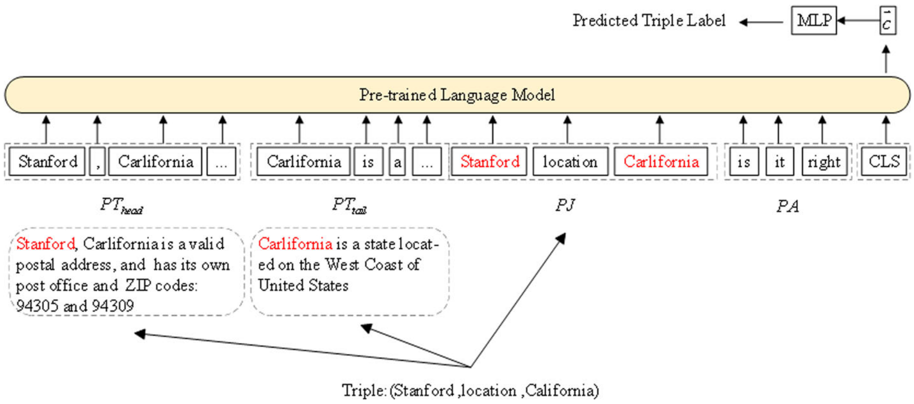
Predicted Triple Label
MLP
C
Pre-trained Language Model
Stanford
,
Carlifornia
...
Carlifornia
is
a
...
Stanford
location
Carlifornia
is
it
right
CLS
PT_head
PT_tail
PJ
PA
Stanford, Carlifornia is a valid postal address, and has its own post office and ZIP codes: 94305 and 94309
Carlifornia is a state located on the West Coast of United States
Triple: (Stanford ,location ,California)

**Fig. 2** Illustration of our PUPKGC model for triple classification. Help triad classification by introducing entity description text and human prior knowledge

[CLS] output of a language model to predict the label of a given triple. On the other hand, this paper uses PU learning to obtain negative triples. The triples classification model receives these triples along with positive samples, and the focus loss is used for training.

The following sections go into great detail about how to obtain negative samples (Sect. 3.3) and design strategies for prompts (Sect. 3.4). In addition, Sect. 3.5 explains how the proposed model is trained.

### 3.3 Negative samples

For KGC, training data consist of positive triples only. Given a positive triple (h, r, t), negative sampling requires sampling one or multiple negative triples to train the discriminant model. To obtain negative samples, existing methods typically involve replacing a part of correct triples at random or manually labeling negative samples. The former method causes positive samples to be mixed with negative samples thus affecting the inference effect, while the latter consumes a lot of time.

We use structural information from the knowledge graph to generate candidate negative examples. The relations of entities and neighboring entities are used for clustering. Two entities can be determined as the same category if they have the same relation to the same or neighboring entities. The method for determining head entities is similar. For training set triplets, we randomly replace their head or tail entities with entities that do not belong to the same class, generating multiple candidate triplet negative samples.

We use the two-step approach in PU learning to generate negative examples and train the model. Specifically, we design a classifier to identify high confidence negative samples. Then mix the generated negative samples with the original positive samples into the model for training. Due to the presence of relevant text information in the candidate triplet negative samples, we utilize existing language models as classifiers. To prevent negative samples recognized by the classifier from being judged as positive samples in the training model, we set the classifier and the model to be trained as the same language model. Considering the use of prompt learning as an auxiliary training tool in the future, for classifiers, we use prompt learning to assist in classification. Please refer to Sect. 3.4 for details. For a set of candidate triplet negative examples, choose the first K triplets with the lowest accuracy as the negative examples.

### 3.4 Prompts

To take advantage of the implicit knowledge within the PLM, each triple is transformed into prompt sentences. For each relation, a hard template is manually designed to represent the semantics of the associated triples. For the triple “[X], position, [Y]”, the [X] and [Y] are first replaced with the exact name of the head and tail entities to get a judgment prompt $\mathrm{PJ}_0$. In this case, $\mathrm{PJ}_0$ is “Stanford position California”. A soft prompt is added to the relation to finally form the more expressive judgment sentence PJ.

To make the inference effect more accurate, text descriptions of the head and tail entities are included in the judgment sentence. Entity definitions or attribute sentences associated with relations are typically used for the text description. To ensure inference accuracy and prevent redundant information interference, the text description is limited to a single sentence that is not overly long. To ensure the accuracy of incoming text description, we use hard prompts instead of soft prompts to form $\mathrm{PT}_{\mathrm{head}}$ and $\mathrm{PT}_{\mathrm{tail}}$. We also add the prompt PA about the task to create the more expressive judgment sentence, which results in the final prompt sentence.

### 3.5 Training

In this paper, the model is trained on a triple set as a triple classification. The negative sample generation method is presented in Sect. 3.3. After comparison, the setting of 1:8 can ensure both low training time and good training effect. Given a triple τ (h, r, t), the classification fraction of the triple can be defined as:

$$s_\tau = \mathrm{Softmax}(Wc) \tag{1}$$

where $c \in Rd$ is the output vector of the input token [CLS], and $W \in R2 \times d$ is a linear neural network. Since the proportion of positive and negative samples is unbalanced, the optimization function in this paper is set as the focus loss.

$$FL(s_\tau) = -\alpha_\tau(1 - s_\tau)^\gamma \log(s_\tau) \tag{2}$$

Parameter $\alpha_\tau$ can suppress the imbalance between the number of positive and negative samples. We make it consistent with the distribution of positive and negative samples in the training dataset. Parameter $\gamma$ can control the difficulty to identify sample number imbalance, and it is set to 2 in this paper to reduce the influence of easily distinguishable samples.

## 4 Experiments

### 4.1 Experimental settings

#### 4.1.1 Datasets

The evaluation in this study employs the WN18RR and FB15k-237 datasets, which are presented in Table 1. The WN18 and FB15k datasets were initially proposed by Bordes et al. [4], but later works [10], Toutanova et al., 2015) revealed that these datasets suffer from test

**Table 1** Statistics of the datasets used in this paper

| dataset | #entity | #relation | #train | #valid | #test |
|---|---|---|---|---|---|
| WN18RR | 40,943 | 11 | 86,835 | 3034 | 3134 |
| FB15k-237 | 14,541 | 237 | 272,115 | 17,535 | 20,466 |

set leakage. To address this issue, the WN18RR and FB15k-237 datasets were created by removing reverse relations. The WN18RR dataset comprises 41 k entities and 11 relations from the WordNet, while the FB15k-237 dataset includes 15 k entities and 237 relations from the Freebase.

For text description, the WN18RR and FB15k-237 datasets provided by the KG-BERT [36] are used in this paper. For the WN18RR dataset, the first sentence is chosen as the text description. For FB15K-237, the first sentence or the sentence related to triples is chosen as the text description.

### 4.2 Evaluation index

The proposed PUPKGC model was evaluated through a link prediction task, following previous studies. Specifically, for each triple (h, r, t) in the test set, the model performed tail entity prediction, which involved determining the likelihood of all possible entities being t given h and r, and then ordering them. Head entity prediction was performed in a similar manner. To assess the model's performance, we used four automatic evaluation indices: mean reciprocal rank (MRR) and Hits@k (H@k) for k ∈ {1,3,10}. MRR was calculated as the mean reciprocal rank of all test triples, while H@k calculated the proportion of correct entities appearing in the top k positions of the ordered rank list. MRR and H@k were reported under filter settings, which ignored the fractions of all known true triples in the training, validation, and test sets.

### 4.3 Hyper-parameter

The encoder is initialized with T5-base. Using an appropriate PLM can further improve performance. Most hyper-parameters are shared across all datasets to avoid specific dataset tuning. The AdamW optimizer is used with linear learning rate attenuation. After comparison, $\alpha = 0.05$ is empirically set. Besides, early stopping is used to balance the training effect and training time.

### 4.4 Main results

The figures reported by scholars are used in this paper for the TransE [4], DKRL [32]. The results of RotatE [26] are from the official GraphVite 4 benchmark. Models with in-batch, pre-batch, and self-negatives are used for the SimKGC [30].

Table 2 shows the performance of our model and baseline model on two datasets. TransE [4], DistMult [35], RotatE [26], and TuckER [2] are typical embedding-based methods. Except for the early TransE and DistMult, all other methods achieved good performance on

**Table 2** Main results for WN18RR and FB15k-237 datasets

| Methods | WN18RR | | | | FB15K-237 | | | |
|---|---|---|---|---|---|---|---|---|
| | MRR | H@1 | H@3 | H@10 | MRR | H@1 | H@3 | H@10 |
| *Embedding-based methods* | | | | | | | | |
| TransE | 24.3 | 4.3 | 44.1 | 53.2 | 27.9 | 19.8 | 37.6 | 44.1 |
| DistMult | 44.4 | 41.2 | 47 | 50.4 | 28.1 | 19.9 | 30.1 | 44.6 |
| RotatE | 47.6 | 42.8 | 49.2 | 57.1 | 33.8 | 24.1 | 37.5 | 53.3 |
| TuckER | 47 | 44.3 | 48.2 | 52.6 | **35.8** | **26.6** | **39.4** | **54.4** |
| *Text-based methods* | | | | | | | | |
| KG-BERT | 21.6 | 4.1 | 30.2 | 52.4 | – | – | – | 42 |
| MTL-KGC | 33.1 | 20.3 | 38.3 | 59.7 | 26.7 | 17.2 | 29.8 | 45.8 |
| StaR | 40.1 | 24.3 | 49.1 | 70.9 | 29.6 | 20.5 | 32.2 | 48.2 |
| SimKGC | **66.6** | **58.7** | 71.7 | 80 | 33.6 | 24.9 | 36.2 | 51.1 |
| PUPKGC | 59.3 | 47.9 | **71.9** | **81.8** | **35.4** | **26.1** | **38.9** | **53.1** |

the FB15K-237 dataset, while performing mediocrely on the WN18RR dataset. Among text-based methods, KG-BERT [36], MTL-KGC [19], and StaR [29] perform poorly and are not as good as traditional embedding-based methods. It should be noted that the simKGC model introduces comparative learning and designs multiple negative sampling categories, achieving significant improvements. Therefore, the performance of simKGC can also confirm our viewpoint that negative sampling restricts the performance of text-based methods. However, simKGC still performs negative sampling through substitution, and there is still the problem of false-negative examples. Our model is highly competitive compared to text-based methods, with six indicators achieving optimal results. Compared to embedding-based methods, our model comprehensively surpasses on the WN18RR dataset and is slightly lower on the FB15K-237 dataset.

FB15k-237 is different from other datasets in that it has a higher chart density (the average degree per entity is about 37) and fewer entities (about 15 k). To perform better, models must learn generable inference rules, not just model text correlation. In this case, an embedding-based method may be advantageous. Consequently, the proposed method can be combined with an embedding-based method. Since it is not the primary focus of this paper, it is viewed as a future work. In addition, scholars point out that many links in the FB15k-237 dataset are unpredictable based on available information [5]. Besides, the entity text description of the FB15k-237 is longer, and it is difficult to find the text description that is suitable for the current entity triple through conventional matching. These three reasons help explain the poor performance of the model in this paper.

The forward transmission of the model is the most expensive part in terms of inference time. For triple inference and judgment, it is often necessary to replace the head and tail entities with other entities for comparison. In this study, we find that replacing triple inference with all entities is time-consuming. Therefore, we suggest using structural information that exists in the KG to limit the inference and comparison scope. Entities are clustered based on their association relations. Finally, for a triple, we only need to replace its head and tail entities with entities in its class for comparative analysis. This method can significantly speed up inference while also improving its accuracy. It should be noted that due to the sparsity of the

WN18RR dataset, there are 294 triples whose entities to be predicted do not belong to the possible classes, and as a result, their link prediction fails.

## 5 Analyses

In this paper, a series of analyses are performed to further understand the proposed model and the KGC task.

### 5.1 Rapid training method

The training time of the model using 4 × 3070 on the FB15K-237 dataset in this paper is about 38 h. There are mainly three reasons for fast training. The ratio of positive to negative samples in this paper is 1:8, which is a relatively low ratio. For description text, an attempt is made to select the entity description associated with the triples; this method can effectively reduce the cost of model training and prevent irrelevant text interference. In addition, the early stopping method is used to avoid additional training and produce the best training results for the model. Experiments show that, for most relations, training can be completed within 8 to 13 periods. However, for relations that frequently occur in the KGs, the training time is usually 8 epochs.

### 5.2 Function of negative sampling

To carefully verify the role of PU learning in the model, a comparative experiment is conducted on the FB15K-237 dataset. The results for the complete model are consistent with those in Sect. 4.2. For comparison, the negative sampling part of the complete model is removed, while the other modules remain unchanged. In terms of this model, the head or tail entities in the triple are randomly replaced with other entities in the training set while ensuring that the newly generated triples do not exist in the training or test sets. These newly generated triples are used as negative samples. The ratio of positive to negative samples is also set to 1:8. The experimental results are shown in Table 3.

In this paper, it is demonstrated that the introduction of PU learning significantly improves the overall performance of the model. For further study and analysis, some of the relations are extracted for illustration, as shown in Table 4.

PU learning can improve the inference effect of this triple on frequent relations in the KGs. However, the improvement is rather evident for relations with average or more frequencies in the KGs. This is considered an outcome of negative sampling. For the triples that frequently appear in the KGs, the probability and proportion of mixed false-negative samples and negative samples produced by random substitution methods are increased, which has a significant

**Table 3** Main results for FB15k-237 dataset

| | MRR | H@1 | H@3 | H@10 |
|---|---|---|---|---|
| PUPKGC | 35.4 | 26.1 | 38.9 | 53.1 |
| Compare Model | 32.9 | 23.3 | 36.8 | 51.8 |

**Table 4** Main results for FB15k-237 dataset of a small number of data sets. The relationships corresponding to the ordinal numbers are listed

| Serial number | Number of test sets | PUPKGC | | | | Compare Model | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | MRR | H@1 | H@3 | H@10 | MRR | H@1 | H@3 | H@10 |
| 1 | 41 | 31.7 | 17.1 | 29.3 | 92.7 | 28.9 | 7.3 | 31.7 | 97.6 |
| 2 | 10 | 18.5 | 10.0 | 10.1 | 50.0 | 8.1 | 0 | 0 | 30.0 |
| 3 | 26 | 9.7 | 3.8 | 11.5 | 23.1 | 8.6 | 3.8 | 7.7 | 15.4 |
| 4 | 436 | 100 | 100 | 100 | 100 | 88.1 | 76.1 | 100 | 100 |
| 5 | 152 | 16.2 | 5.3 | 17.1 | 36.2 | 16.9 | 8.6 | 13.2 | 36.8 |
| 6 | 91 | 16.8 | 11.0 | 19.8 | 23.1 | 19.6 | 11.0 | 26.4 | 30.8 |
| 7 | 15 | 8.2 | 6.7 | 6.7 | 6.7 | 2.4 | 0 | 0 | 6.7 |
| 8 | 6 | 12.5 | 0 | 0 | 50.0 | 11.8 | 0 | 0 | 50.0 |
| 9 | 323 | 69.6 | 56.0 | 76.8 | 98.5 | 57.8 | 47.4 | 61.6 | 83.0 |
| 10 | 24 | 75.5 | 75.0 | 75.0 | 75.0 | 77.5 | 75.0 | 79.2 | 83.3 |

influence on the quality of the training dataset. Therefore, the introduction of PU learning to help negative sampling significantly enhances the impact of low-frequency sampling. This demonstrates in full the rationality of this paper's starting point.

### 5.3 Effectiveness of auxiliary inference

Prior embedding-based methods have also achieved some success in inference. This fully shows the role of graph structural information in KGC. Therefore, we use the structural information of the KG for clustering. For KGs with rich relational information, such as the FB15K-237 dataset, clustering information is used for inference, as detailed in Sect. 4.2.

In order to explore the role of auxiliary inference modules, we designed a comparative experiment. Specifically, we removed the auxiliary reasoning part from the original model and compared it with the original model. For previous methods, the number of times required for comparing and judging a single triplet is determined by the following formula.

$$N = \frac{N_{\text{entity}} * N_{\text{test}} - N_{\text{all}}}{N_{\text{test}}} \tag{3}$$

$N_{entity}$ is the number of entities in the knowledge graph dataset. $N_{test}$ is the number of triples in the knowledge graph test dataset. $N_{all}$ is the number of all triples in the knowledge graph dataset. $N$ is the number of times an average single triplet needs to be compared and judged. We substituted it into Table 1 and found that for the FB15K-237 dataset, the average number of times a single triplet needs to be judged and compared is about 14,526. After introducing the auxiliary reasoning module, it was found through testing that this frequency can be reduced to 524.

Meanwhile, it should be noted that another 15 test triplets cannot be linked to the correct entity. This number is not worth mentioning compared to the 20,466 data in the test set. Therefore, our auxiliary inference module can save 96% of time on the FB15K-237 dataset. This fully proves the effectiveness of our module.

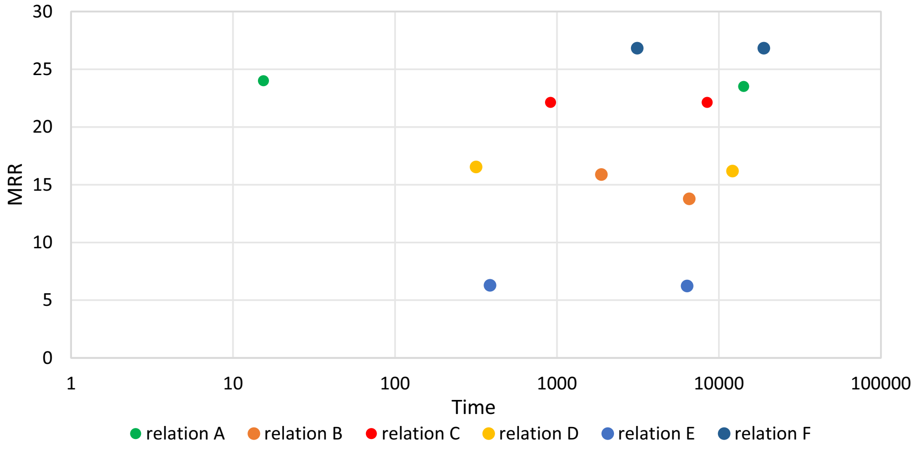

**Fig. 3** The influence of auxiliary reasoning on partial relations

We extracted some relationships for further comparative analysis, and the results are shown in Fig. 3. The x-coordinate is the inference time, in seconds, and the y-coordinate is the MRR value. A set of points with the same color is a representation of the same triples in the two models. A set of points with the same color is the experimental result of the same relationship in two models. On the left is the experimental results of the model that introduces structural information for auxiliary reasoning, while on the right is the experimental results of the original model. From the figure, it can be seen that the introduction of structural information greatly improves the inference speed. The inference accuracy of the model has also been improved, and the effect is more pronounced for relationships with lower frequencies in KG. This is mainly the result of eliminating the influence of false-positive entities during the model training process. This fully demonstrates the rich informational content of the KG structure and the efficiency of the module.

## 6 Conclusion and future works

With the rapid development of PLMs, some PLM-based KGC models have been proposed. However, there is still a performance gap between these models and the SOTA KGE model. In this work, we identify that the key factor of poor performance lies in negative sample collection. To address this issue, we introduce PU learning to help generate negative samples, propose a new PLM-based KGC model, and verify the influence of negative sample collection on inference. Experimental results show that the proposed model produces better results than the previous text-based methods. In a future work, we plan to improve the module on negative sample collection and explore the potential for expanding the usage of structural information.

**Acknowledgements** This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

**Author contributions** DL did conceptualization, methodology, validation, writing—review & editing. WJ done data preparation, software, writing—original draft, writing—review & editing. LB contributed to conceptualization, methodology, validation, investigation. SQ performed conceptualization, methodology, validation, investigation, supervision, and project administration.

**Funding** No funding was received for conducting this study.

**Code availability** Code that supports the findings of this study and data extracted for training the classification algorithms are available from the corresponding author upon reasonable request.

## Declarations

**Competing interests** The authors declare no competing interests.

**Conflict of interest** The authors declare that they have no conflict of interest.

**Ethics approval** We declare that this submission follows the policies as outlined in the Guide for Authors. The current research involves no Human Participants and/or Animals.

## References

1. 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: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007+ ASWC 2007, Busan, Korea, 2007. Proceedings. Springer, pp. 722–735.
2. Balaevi I, Allen C, Hospedales TM (2019) TuckER: tensor factorization for knowledge graph completion.
3. 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, 1247–1250.
4. 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.
5. Cao Y, Ji X, Lv X, Li J, Wen Y, Zhang H, (2021) Are missing links predictable? An inferential benchmark for knowledge graph completion. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 6855–6865.
6. Chao L, He J, Wang T, Chu W (2020) Pairre: knowledge graph embeddings via paired relation vectors. arXiv preprint arXiv:2011.03798.
7. Chen X, Xie X, Zhang N, Yan J, Deng S, Tan C, Huang F, Si L, Chen H (2021) Adaprompt: adaptive prompt-based finetuning for relation extraction. arXiv preprint arXiv:2104.07650.
8. Cui L, Wu Y, Liu J, Yang S, Zhang Y (2021) Template-based named entity recognition using BART. Find Assoc Comput Ling: ACL-IJCNLP 2021:1835–1845
9. Daza D, Cochez M, Groth P (2021) Inductive entity representations from text via link prediction. Proc Web Conf 2021:798–808
10. Dettmers T, Minervini P, Stenetorp P, Riedel S, (2018) Convolutional 2d knowledge graph embeddings, Proceedings of the AAAI conference on artificial intelligence.
11. Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
12. Dong X, Gabrilovich E, Heitz G, Horn W, Lao N, Murphy K, Strohmann T, Sun S, Zhang W, (2014) Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 601–610.
13. Gao T, Fisch A, Chen D (2020) Making pre-trained language models better few-shot learners. arXiv preprint arXiv:2012.15723.
14. Hambardzumyan K, Khachatrian H, May J (2021) Warp: word-level adversarial reprogramming. arXiv preprint arXiv:2101.00121.
15. Han X, Zhao W, Ding N, Liu Z, Sun M (2022) Ptr: prompt tuning with rules for text classification. AI Open 3:182–192
16. Hao Y, Zhang Y, Liu K, He S, Liu Z, Wu H, Zhao J, (2017) An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 1: 221–231.
17. He F, Liu T, Webb GI, Tao D (2018) Instance-dependent pu learning by bayesian optimal relabeling. arXiv preprint arXiv:1808.02180.
18. Jiang Z, Xu FF, Araki J, Neubig G (2020) How can we know what language models know? Trans Assoc Comput Ling 8:423–438

19. Kim B, Hong T, Ko Y, Seo J, (2020) Multi-task learning for knowledge graph completion with pre-trained language models. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 1737–1743.
20. Liu B, Liu Q, Xiao Y (2022) A new method for positive and unlabeled learning with privileged information. Appl Intell 52:2465–2479
21. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. Adv Neural Inf Process Syst, 26.
22. Pennington J, Socher R, Manning CD, (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 1532–1543.
23. Radford A, Narasimhan K, Salimans T, Sutskever I (2018) Improving language understanding by generative pre-training.
24. Sha X, Sun Z, Zhang J (2021) Hierarchical attentive knowledge graph embedding for personalized recommendation. Electron Commer Res Appl 48:101071
25. Suchanek FM, Kasneci G, Weikum G, (2007) Yago: a core of semantic knowledge. In: Proceedings of the 16th international conference on World Wide Web, pp. 697–706.
26. Sun Z, Deng Z-H, Nie J-Y, Tang J (2019) Rotate: knowledge graph embedding by relational rotation in complex space. arXiv preprint arXiv:1902.10197.
27. Toutanova K, Chen D, Pantel P, Poon H, Choudhury P, Gamon M, (2015) Representing text for joint embedding of text and knowledge bases. In: Proceedings of the 2015 conference on empirical methods in natural language processing, pp. 1499–1509.
28. Trouillon T, Welbl J, Riedel S, Gaussier É, Bouchard G, (2016) Complex embeddings for simple link prediction. In: International conference on machine learning. PMLR, pp. 2071–2080.
29. Wang B, Shen T, Long G, Zhou T, Wang Y, Chang Y (2021) Structure-augmented text representation learning for efficient knowledge graph completion. Proc Web Conf 2021:1737–1748
30. Wang L, Zhao W, Wei Z, Liu J, (2022) 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), 4281–4294.
31. Wang Q, Mao Z, Wang B, Guo L (2017) Knowledge graph embedding: a survey of approaches and applications. IEEE Trans Knowl Data Eng 29:2724–2743
32. Wang X, Gao T, Zhu Z, Zhang Z, Liu Z, Li J, Tang J (2021) KEPLER: a unified model for knowledge embedding and pre-trained language representation. Trans Assoc Comput Ling 9:176–194
33. Wang Z, Zhang J, Feng J, Chen Z, (2014) Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI conference on artificial intelligence.
34. Xie R, Liu Z, Jia J, Luan H, Sun M, (2016) Representation learning of knowledge graphs with entity descriptions. In: Proceedings of the AAAI Conference on Artificial Intelligence.
35. Yang B, Yih WT, He X, Gao J, Deng L, (2014) Embedding entities and relations for learning and inference in knowledge bases. In: International Conference on Learning Representations.
36. Yao L, Mao C, Luo Y (2019) KG-BERT: BERT for knowledge graph completion. arXiv preprint arXiv:1909.03193.
37. Zhang F, Yuan NJ, Lian D, Xie X, Ma W-Y, (2016) Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 353–362.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Li Duan** was born in Jian, JiangXi, China, in 1976. He received the Ph.D. degree from Huazhong University of Science and Technology, Hubei Whan, China, in 2009. He is currently a professor with the Naval University of Engineering. His current research interests include data fusion, knowledge engineer and military artificial intelligence.

**Jing Wang** was born in XianTao, Hubei, China, in 1998. He received B.S.degree in Engineering of fire command and control from Naval University of Engineering, PLA, Wuhan, China, in 2019. He is currently working toward the M.Sc. degree in Knowledge engineering and intelligent systems from Naval University of Engineering, PLA. His currently research interests include knowledge graph and natural language processing.

**Bing Luo** was born in Xiangtan, Hunan, China, in 1979. He received a master’s degree from the National University of Defense Technology, Changsha, China, in 2003. He is currently an associate professor with the Naval University of Engineering. His current research interests include command information system, military knowledge engineering and multimodal semantic understanding.

**Qiao Sun** was born in Wenzhou, Zhejiang, China, in 1989. He received the Ph.D. degree from the National University of Defense Technology, Changsha, China, in 2016. He is currently a lecturer with the Naval University of Engineering, and also working as a postdoctoral researcher with Zhejiang University, Hangzhou, China. His current research interests include representation learning, data mining and multimodal semantic retrieval.