information

Article

Simple Knowledge Graph Completion Model Based on Differential Negative Sampling and Prompt Learning

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

[1] College of Electronic Engineering, Naval University of Engineering, Wuhan 430033, China
[2] College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China
* Correspondence: m21181005@nue.edu.cn

Abstract: Knowledge graphs (KGs) serve as a crucial resource for numerous artificial intelligence tasks, significantly contributing to the advancement of the AI field. However, the incompleteness of existing KGs hinders their effectiveness in practical applications. Consequently, researchers have proposed the task of KG completion. Currently, embedding-based techniques dominate the field as they leverage the structural information within KGs to infer and complete missing parts. Nonetheless, these methods exhibit limitations. They are limited by the quality and quantity of structural information and are unable to handle the missing entities in the original KG. To overcome these challenges, researchers have attempted to integrate pretrained language models and textual data to perform KG completion. This approach utilizes the definition statements and description text of entities within KGs. The goal is to compensate for the latent connections that are difficult for traditional methods to obtain. However, text-based methods still lag behind embedding-based models in terms of performance. Our analysis reveals that the critical issue lies in the selection process of negative samples. In order to enhance the performance of the text-based methods, various types of negative sampling methods are employed in this study. We introduced prompt learning to fill the gap between the pre-training language model and the knowledge graph completion task, and to improve the model reasoning level. Simultaneously, a ranking strategy based on KG structural information is proposed to utilize KG structured data to assist reasoning. The experiment results demonstrate that our model exhibits strong competitiveness and outstanding inference speed. By fully exploiting the internal structural information of KGs and external relevant descriptive text resources, we successfully elevate the performance levels of KG completion tasks across various metrics.

Keywords: natural language processing; knowledge graph completion; prompt learning; positive unlabeled learning

check for updates

Citation: Duan, L.; Wang, J.; Luo, B.; Sun, Q. Simple Knowledge Graph Completion Model Based on Differential Negative Sampling and Prompt Learning. *Information* **2023**, *14*, 450. https://doi.org/10.3390/info14080450

Academic Editors: Tudor Groza and Ralf Krestel

Received: 29 May 2023
Revised: 4 August 2023
Accepted: 4 August 2023
Published: 9 August 2023

1. Introduction

Large-scale knowledge graphs such as FreeBase [1], YAGO [2], and DBpedia [3] have been instrumental in supporting various artificial intelligence systems. These systems include semantic search [4], recommendation systems [5] and question-answering systems [6]. They have been widely applied in numerous domains, such as finance and healthcare, benefiting human life and society.

Due to technical problems and explosive information growth, existing KGs often suffer from data incompleteness. This issue has inspired the task of knowledge graph completion (KGC), which aims to evaluate the plausibility of potential triples and enrich the KG. Many studies have focused on KGC, with one common approach being knowledge graph embedding (KGE). KGE maps entities and relations to low-dimensional vectors and evaluates triples using these vectors [7]. Typical models include TransE [8], TransH [9], RotatE [10] and TuckER [11]. Text-based methods [12–14] utilize available textual information uses the available text information of KGC to learn semantic relations, so as to complete the knowledge graph. The fundamental difference between these two methods can be seen in Figure 1. Intuitively, text-based methods should outperform embedding-based methods due to the additional information they incorporate. However, experiments on some datasets show that text-based methods lag behind structure-based methods.

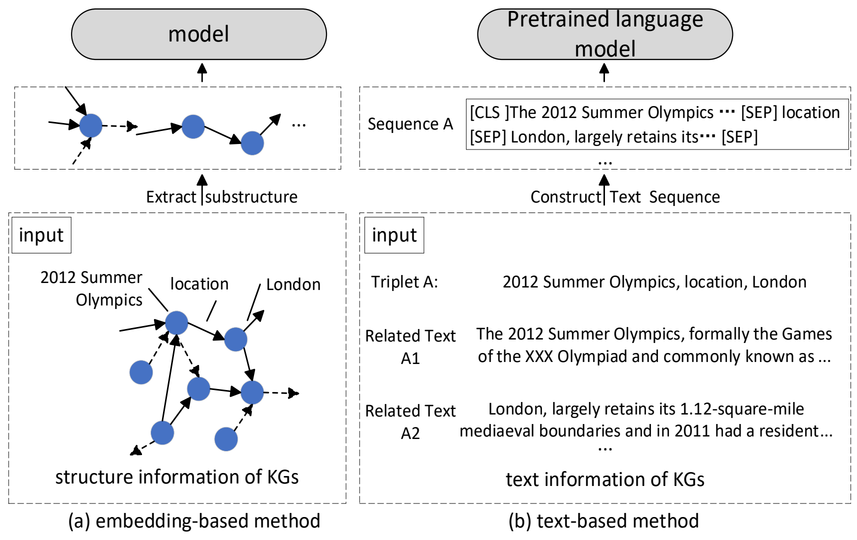

model
Pretrained language model
Sequence A
[CLS]The 2012 Summer Olympics ··· [SEP] location [SEP] London, largely retains its··· [SEP]
Extract substructure
Construct Text Sequence
input
2012 Summer Olympics
location
London
structure information of KGs
Triplet A:
2012 Summer Olympics, location, London
Related Text A1
The 2012 Summer Olympics, formally the Games of the XXX Olympiad and commonly known as ...
Related Text A2
London, largely retains its 1.12-square-mile mediaeval boundaries and in 2011 had a resident...
text information of KGs
(a) embedding-based method
(b) text-based method

Figure 1. Different KGC methods. (**a**) A traditional embedding-based method learns the embedded representation of entities and relations in the knowledge graph through the structural information of the knowledge graph, so as to complete the KG. This approach suffers from the lack of information about the structure of the knowledge graph. (**b**) A text-based method overcomes the constraints of structural information by using text information in the knowledge graph and pretrained language models to make triplet judgment to complete the KG. However, due to the similarity of the text, the inference effect in some relations needs to be strengthened.

We found that the introduction of text information increases the similarity between entities of the same category. For example, in the FB15K-237 dataset, the relation "/people/person/languages" has tail entities as languages, and their entity descriptions show a certain similarity, which interferes with the model's inference effect. Therefore, we believe that this similarity increases the difficulty of model learning and is the reason for the poor performance of text-based methods.

It is necessary to use a proper negative sampling method to help the model distinguish between similar entities. In order to improve the effect of negative sampling, we divided the relations in the KGs into three types according to the structure of the KGs. We chose different negative sampling methods based on different types of relations and constructed appropriate negative samples to improve model learning effectiveness.

Recently, prompt learning has received more and more scholars' attention because of its superior performance [15]. At the same time, it has achieved good performance in many natural language processing (NLP) downstream tasks. To further improve the training effectiveness of the model, we introduced prompt learning. In this process, the triples of positive and negative samples and their related entity descriptions were filled into templates. These were then input into a pretrained language model (PLM) for training after introducing labels. To optimize the training results, we incorporated Focal loss. This adjusts the impact of the positive and negative sample ratio imbalance and the influence of hard/easy-to-distinguish samples on the model. The trained model was then used to predict the plausibility of triples. This method exhibits strong performance in multiple KGC tasks. The contributions of this paper are summarized as follows:

- We proposed a KGC method based on pretrained language models, which combines KG structure to solve the negative sampling problem;
- Results from multiple benchmark datasets demonstrate that our method achieves competitive performance in link prediction tasks;

- Compared to models with similar accuracy, our model significantly reduces inference time.

2. Related works

2.1. Knowledge Graph Completion

Knowledge graph completion aims to model multi-relational data to help fill in missing information in existing knowledge graphs. Traditional approaches typically employ structural information from knowledge graphs for reasoning, such as TransE [8] and TransH [9]. They treat triples (h, r, t) as specific relation transformations from head entity h to tail entity t. Complex [16] introduces multiple embeddings to enhance model expressiveness, while RotatE [10] simulates triple relation rotations in complex space. Complex relation patterns are encoded by some researchers using two vectors per relation and adaptively adjusting margin parameters in the loss function [17]. Recently, additional textual information has been utilized to assist knowledge graph completion. DKRL [18] encodes text using CNN [19] and continuous bag of words(CBOW). KG-BERT [12], StAR [13], BLP [20] and simKGC [14] computing entity embeddings using pretrained language models. Although these methods improve performance, they still underperform embedding-based approaches on some datasets. Some scholars have used descriptive text and language models to derive knowledge embeddings. This approach not only enriches the representation of long-tail entities, but also solves the problem based on prior description methods. It achieves better performance than previous embedding-based models [21].

2.2. Pretrained Language Models

Pretrained language models, having demonstrated immense potential in various NLP tasks [22], are typically pretrained on large-scale corpora, thereby storing vast amounts of general knowledge. Most of these models are derived from the Transformer design [23], which comprises encoder and decoder modules enhanced by a self-attention mechanism. Depending on the architectural structure, pretrained language models are categorized into three groups: encoder-only models, encoder-decoder models and decoder-only models.

Encoder-only models, such as BERT [24], ALBERT [25] and RoBERTa [26], utilize the encoder to comprehend the relations between words within a sentence. An additional prediction head is usually required for these models to resolve downstream tasks. They are particularly effective for tasks necessitating a comprehensive understanding of an entire sentence, such as text classification [27] and named entity recognition [28].

Encoder-decoder models, on the other hand, incorporate both the encoder and decoder modules. The encoder module encodes the input sentence into a hidden space, while the decoder generates the target output text. These models, including T5 [29], UL2 [30] and ST-MoE [31], offer more flexible training strategies.

Decoder-only large language models exclusively use the decoder module to generate the target output text. Their training paradigm involves predicting the subsequent word in a sentence. These large-scale models can generally perform downstream tasks from a few examples or simple instructions without the need for additional prediction heads or finetuning [32]. Many state-of-the-art Pretrained language models (e.g., Chat-GPT [33] and GPT-4) follow the decoder-only architecture.

2.3. Prompt Learning

Prompt learning, unlike conventional supervised learning, trains language models based on direct text probability modeling. In this approach, models receive input x and predict output y. To adapt these models for predictive tasks, templates transform the original input x into a text string prompt with fillable slots. These slots are then populated to obtain the final string Pt, which is inputted into the pretrained language model to generate the output y. This framework is powerful and attractive for several reasons. It allows for the pre-training of language models on vast amounts of raw text. By defining a new prompt function, the model can perform few-shot or even zero-shot learning. This adaptability enables the model to adjust to new scenarios with minimal or no labeled data. Prompt learning has achieved impressive results in natural language processing tasks such as text classification [34], relation extraction [35], named entity recognition [36] and question-answering systems [37].

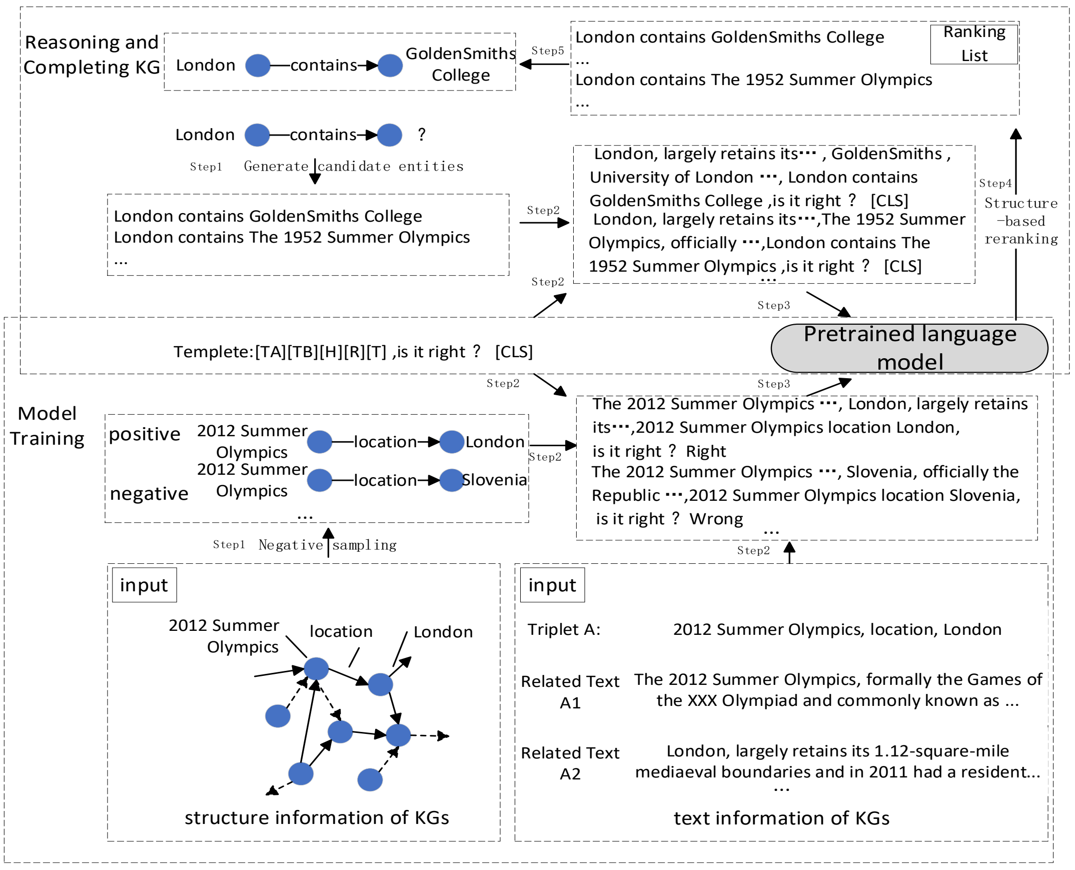
2.4. Positive Unlabeled Learning

Positive unlabeled (PU) learning is based on scenarios where researchers can access only positive examples and unlabeled data. This scenario has gained increasing attention as it naturally occurs in applications such as medical diagnosis and knowledge graph completion. The objective of PU learning is the same as general binary classification: training a classifier capable of classifying based on target attributes. Most methods can be divided into three categories: two-step approaches [38], biased learning [39] and class prior integration methods. Two-step techniques involve two steps: (1) identifying high-confidence negative examples, and (2) learning based on labeled positive and high-confidence negative examples. Biased learning methods treat unlabeled data as negative examples with class label noise. Class prior integration includes postprocessing, preprocessing and model modification methods. Its idea is to introduce class priors to modify traditional learning approaches.

3. Methods

3.1. Framework

We propose a new KGC completion method. This method uses the structural information in KGs to solve the negative sampling problem and introduces prompt learning to improve the model training effect. The methods complete the knowledge graph by learning the text information in the knowledge graph, combining the implicit knowledge in PLM and the structural information in KGs. The framework of the method is shown in Figure 2.



Reasoning and Completing KG
London contains GoldenSmiths College
Step5
London contains GoldenSmiths College
...
London contains The 1952 Summer Olympics
...
Ranking List
London contains ?
Step1 Generate candidate entities
London contains GoldenSmiths College
London contains The 1952 Summer Olympics
...
Step2
London, largely retains its··· , GoldenSmiths , University of London ···, London contains GoldenSmiths College ,is it right ? [CLS]
London, largely retains its···,The 1952 Summer Olympics, officially ···,London contains The 1952 Summer Olympics ,is it right ? [CLS]
...
Step4 Structure-based reranking
Step3
Templete:[TA][TB][H][R][T] ,is it right ? [CLS]
Pretrained language model
Model Training
positive 2012 Summer Olympics location London
negative 2012 Summer Olympics location Slovenia
...
Step1 Negative sampling
The 2012 Summer Olympics ···, London, largely retains its···,2012 Summer Olympics location London, is it right ? Right
The 2012 Summer Olympics ···, Slovenia, officially the Republic ···,2012 Summer Olympics location Slovenia, is it right ? Wrong
...
input
2012 Summer Olympics
location
London
structure information of KGs
input
Triplet A: 2012 Summer Olympics, location, London
Related Text A1 The 2012 Summer Olympics, formally the Games of the XXX Olympiad and commonly known as ...
Related Text A2 London, largely retains its 1.12-square-mile mediaeval boundaries and in 2011 had a resident...
...
text information of KGs

Figure 2. Knowledge graph completion method framework.

In the training stage, we initially leverage structural information of the knowledge graph to generate negative samples. Subsequently, we merged positive and negative samples, text information and set templates to generate the necessary training data for prompt learning. These data are then incorporated into a pretrained language model for training.

During the inference phase, we commence by generating a collection of candidate entities to be linked in the missing triplets. The candidate entity set, along with the missing triples and text information, are imported into the template for processing. These processed inputs are then fed into the trained model to yield scores. To enhance the reasoning effect, we utilize the structural information of the knowledge graph to perform re-ranking. Finally, the triplet with the highest score is integrated into the knowledge graph for completion.

The following sections go into great detail about how to obtain negative samples (Section 3.2) and design strategies for prompts (Section 3.3). In addition, Section 3.4 explains how the proposed model is trained, and Section 3.5 introduces the reranking method based on knowledge graph structure information.

3.2. Negative Samples

In knowledge graph completion, training data consist solely of positive triplets. Given a positive triplet (h, r, t), negative sampling requires sampling one or more negative triplets for training the discriminative model. Existing methods primarily involve randomly replacing parts of the correct triplet to form negative examples or manually annotating negative examples. These approaches do not consider the influence of different relation types on negative sampling and the introduced negative examples cannot effectively help discriminate between entities within the tail entity's class. Our method significantly improves training results without incurring substantial computational overhead.

3.2.1. Relation Category Definition

Because the relation types of triples are different, the corresponding distribution of error triples is also different. Therefore, it is not appropriate to use the same negative sampling method for all triples. Therefore, we divide the triples in the knowledge graph into three categories based on the structural information of the triples: to-one relations, to-class relations and to-many relations.

Definition 1. *To-one relations refers to the relation in the knowledge graph where a given related head entity has only one corresponding tail entity.*

For example, 'people/person/place_of_birth' is a typical to-one relation, for any given head entity (person), there is only one corresponding tail entity (place).

Definition 2. *To-class relation refers to the relation in the knowledge graph where, for a given related head entity, the corresponding tail entities are all entities under a category.*

"/soccer/football_team/current_roster./soccer/football_roster_position/position" refers to the football position corresponding to the football club, which is a typical to-class relation. For a given header entity, the corresponding tail entity is the tail entity in an entire category.

Definition 3. *To-many relation refers to the relation in the knowledge graph that for a given related head entity, the corresponding tail entities belongs to a category, but does not include all entities under the category.*

'language/human_language/countries_spoken_in' is a to-many relation, that is, its tail entities all belong to a category (country), but for a given head entity (language), there may be more than one tail entity, but not all entities under that category.

3.2.2. Negative Samples Strategy

For negative sampling of to-one relations, the focus should be on finding the correct entity in the possible class for a given relation. For instance, in the "/location/country/capital" relation, which is a typical to-one relation, connecting a given correct head entity to a specific city is relatively easy, while linking it to other entities like "dollar" is absurd. Identifying the correct city, however, is challenging. Therefore, for this type of negative sampling, the emphasis should be on replacing the tail entity with entities from the possible class. The specific details are available in Algorithm 1.

Algorithm 1: Generate negative samples for to-one relations

```
Input: To-one triplet set T, entity set E, validation triplet set T′, integer k.
1.  Initialize a new triplet set N
2.  for each triplet (head_entity, relation, tail_entity) in T do
3.      num ← 0
4.      Find the entity set E′ where the tail_entity belongs form E
5.      while num ≤ k do
6.          Randomly select an entity new_tail_entity from E′
7.          generating a new triplet (head_entity, relation, new_tail_entity)
8.          if the new triple not in T and t′ is not in T′ then
9.              Add the new triple to N
10.             num ← num + 1
11.         end if
12.     end while
13. end for
Output: The negative samples set N.
```

For negative sampling of to-class relations, the focus should be on learning the relation between the entity and the entities within the class. Hence, for this type of negative sampling, the emphasis should be on entities outside the possible class. The specific details are available in Algorithm 2.

Algorithm 2: Generate negative samples for to-class relations

```
Input: To-class triplet set T, entity set E, validation triplet set T′, integer k.
1.  Initialize a new triplet set N
2.  for each triplet (head_entity, relation, tail_entity) in T do
3.      num ← 0
4.      Find the entity set E′ where the tail_entity not belongs form E
5.      while num ≤ k do
6.          Randomly select an entity new_tail_entity from E′
7.          generating a new triplet (head_entity, relation, new_tail_entity)
8.          if the new triple not in T and t′ is not in T′ then
9.              Add the new triple to N
10.             num ← num + 1
11.         end if
12.     end while
13. end for
Output: The negative samples set N.
```

For the negative sampling of to-many relations, the distinction between the target entity and the class and the difference from entities outside the class should be learned at the same time. For this type of entity, we randomly replace the tail entity with multiple entities in the knowledge graph. In addition, we incorporate them into a pretrained language model for inference and select the K triplet with the lowest correctness as a negative example. The above methods can be regarded as a two-step technique in PU learning. The specific details are available in Algorithm 3.

Algorithm 3: Generate negative samples for to-many relations

Input: To-many triplet set T, entity set E, validation triplet set T′, integer j, integer k, Triplet classification model M.

```
1.  Initialize a new triplet set N
2.  Initialize a new triplet set N2
3.  Initialize a new triplet set N3
4.  Training model M with To-many triplet set T and validation triplet set T′
5.  for each triplet (head_entity, relation, tail_entity) in T do
6.      num ← 0
7.      while num ≤ j do
8.          Randomly select an entity new_tail_entity from E
9.          generating a new triplet (head_entity, relation, new_tail_entity)
10.         if the new triple not in T and t′ is not in T′ then
11.             Add the new triple to N
12.             num ← num + 1
13.         end if
14.     end while
15.     for each triplet (head_entity, relation, new_tail_entity) in T do
16.         Eval (head_entity, relation, new_tail_entity) with Traing model M to get eval_soccer
17.         Add (new_tail_entity, eval_soccer) to N2
18.     end for
19.     Sort N2 by eval_soccer
20.     for i = 1 to k do
21.         Add the (head_entity, relation, new_tail_entity) to N3
22.     N ← []
23.     N2 ← []
24. end for
```

Output: The negative samples set N3.

3.3. 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], language, [Y]", the [X] and [Y] are first replaced with the exact name of the head and tail entities to get a judgment prompt PJ_0. In this case, PJ_0 is "Mukri language Hindi Language". 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 PT_{head} and PT_{tail}. We also add an assist prompt PA about the task to create the more expressive judgment sentence, which results in the final prompt sentence.

The process of prompts design is shown in Figure 3, and we use (Mukri, language, Hindi Language) as an example to illustrate it.

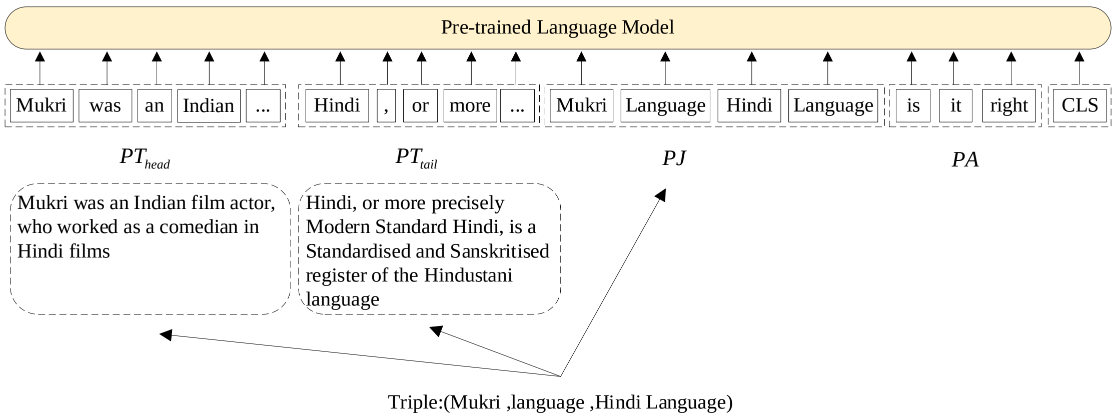

Pre-trained Language Model
Mukri
was
an
Indian
...
Hindi
,
or
more
...
Mukri
Language
Hindi
Language
is
it
right
CLS
PT_{head}
PT_{tail}
PJ
PA
Mukri was an Indian film actor, who worked as a comedian in Hindi films
Hindi, or more precisely Modern Standard Hindi, is a Standardised and Sanskritised register of the Hindustani language
Triple:(Mukri ,language ,Hindi Language)

Figure 3. A schematic diagram of the process of generating prompts.

3.4. Training

In this paper, the model is trained on a triple set as a triple classification. The negative sample generation method is presented in Section 3.3. After comparison, the setting of 1:10 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

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

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 α_τ 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 γ 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.

In order to accelerate training speed and prevent overfitting, we introduce the early stopping. We set the patient of the early stop method to 7 to ensure that the model is fully trained.

3.5. Structure-Based Reranking

Knowledge graphs tend to be spatially relevant. For some relations, there is a multi-hop association between the head-tail entities. The text-based knowledge graph completion model is good at capturing semantic correlation, but not structural correlation. We propose a simple reordering strategy based on the structure of the knowledge graph: for a set of triplet τ (h, r, t) model scores that exceed the threshold and the tail entity is in the n-hop neighbor of the head entity, the score of the tail entity is increased by $\alpha \geq 0$. Therefore, for a given τ (h, r, t_i), the reasoning score after re ranking is

$$\underset{t_i}{s'_\tau} = \underset{t_i}{s_\tau} + \alpha(t_i \in \varepsilon_k(h), s_\tau \geq s_0) \tag{3}$$

4. Experiments

4.1. Evaluation Protocol

The task of link prediction in KGs involves predicting the missing triples when the KG is incomplete. Specifically, for each triple (h, r, t) in the test set, the model performed tail entity prediction. This process involves determining the likelihood of all possible entities being t given h and r, and then ordering them. 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.

Regarding inference time, the most expensive component is the forward pass of the model. Inference and evaluation of triplets typically require replacing head and tail entities with other entities for comparison. We find that replacing triplets with all entities for inference is highly time-consuming. Therefore, we propose leveraging existing structural information within knowledge graphs to reduce the scope of inference and comparison. We cluster entities based on their associated relations. Ultimately, for a given triplet, we only need to replace its head and tail entities with those within their respective classes for comparative analysis. This approach not only substantially accelerates the inference process but also enhances its accuracy.

To assess the model's performance, we used four automatic evaluation indices: mean reciprocal rank (MRR) and Hits@k (H@k) for k$\in$\{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.

4.2. Experimental Settings

4.2.1. 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, Usunier, Garcia-Duran, Weston and Yakhnenko [8], but later works [40,41] revealed that these datasets suffer from test set leakage. To address this issue, the WN18RR and FB15k-237 datasets were created by removing reverse relations. The WN18RR dataset comprises 41k entities and 11 relations from the WordNet, while the FB15k-237 dataset includes 15k entities and 237 relations from the Freebase.

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

For text description, the WN18RR and FB15k-237 datasets provided by the KG-BERT [12] are used in this paper. The Wikidata5M dataset already contains descriptions of all entities and relations. 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.

4.2.2. 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. In addition, early stopping is used to balance the training effect and training time.

4.3. Main Results

In this study, we employed the numerical values reported by Wang et al. [42] for TransE and DKRL, while the results for RotatE were obtained from the official GraphVite 4 benchmark. For SimKGC, we utilized a model incorporating batch-wise negative, pre-batch negative and self-negative samples. The results for C-LMKE are cited from the literature reports.

In Table 2, we can clearly see that only a small number of text-based methods outperform structure-based methods on the WN18RR or FB15K-237 datasets. Regardless of our approach, only C-LMKE outperforms the structure-based approach on both datasets. This validates the necessity of our research.

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
DisMult	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
C-LMKE	59.8	48.0	67.5	80.6	40.4	32.4	43.9	55.6
SimPKGC	62.6	56.2	65.1	76.3	41.2	33.7	44.3	55.4

In the WN18RR dataset, all four indicators of our proposed model are in the top three, and two of them are in second place. On the data set FB15K-237, all three indexes are better than other models, and the remaining index reaches the second place. At the same time, our models all outperform the structure-based methods. It can be seen that our model has achieved competitive performance in link prediction tasks.

4.4. Analyses

On the WN18RR dataset, our model outperforms embedding-based models and is highly competitive with text-based models. On the FB15K-237 dataset, our model exceeds most of the baseline models and achieves the best on three metrics. In summary, our model is highly competitive. Especially in the FB15K-237 dataset, where most text-based models perform poorly, our model achieves outstanding performance. This definitively shows that the text-based model performs better than the embedding-based model, and also verifies the rationality of our starting point. The training time of the model using 4×3070 on the FB15K-237 dataset in this paper is about 44 h. Compared to other text-based methods, it takes very little time and fully demonstrates the superiority of our model.

We analyzed the gap between the optimal metrics of our model indicators on the WN18RR dataset. We believe there are the following reasons. First of all, due to the sparsity of the WN18RR dataset, some triplets' target entities are not present in the possible classes, leading to the failure of link prediction for these triplets. Secondly, because some of the entity text descriptions in the WN18RR dataset are difficult to intuitively reflect the relation between entities, this also leads to a poor reasoning effect on some relations.

In order to explore the role of additional modules within the model and better dissect the proposed model, we conducted additional analysis.

4.4.1. Rapid Training Method

The training time of the model using 4×3070 on the FB15K-237 dataset in this paper is about 44 h. There are mainly three reasons for fast training. The ratio of positive to negative samples in this paper is 1:10, 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.

In Figure 4, solid and dashed lines of the same color represent the training accuracy and verification accuracy of the same relation, respectively. Figure 4 shows the number of early stopping occurrences and accuracy of some relations in the FB15K-237 dataset. The figure shows that, for most relations, training can be completed within 8 to 11 periods. Meanwhile, it shows that as the number of experiments increases, the model undergoes overfitting after a certain point. This fully demonstrates that introducing the early stopping method can ensure both training speed and accuracy.

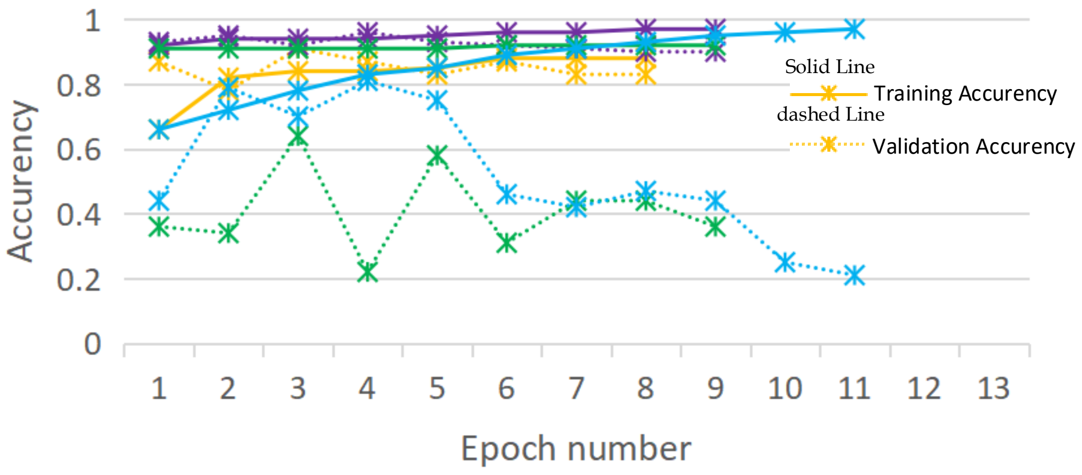


Figure 4. Trianing epoch number of the model under FB15K-237 partial relation using the early stopping method. Five different colors correspond to five training data of our five relationships, and the selection of relationships is random.

4.4.2. Function of Negative Sampling

In order to rigorously investigate the role of our proposed negative sampling method in the model, we conducted comparative experiments on the FB15K-237 dataset. The data results of the complete model are consistent with those of Section 4.2. For comparison, we remove the negative sampling function from the complete model, while the other functions remain unchanged to form a new model (SimPKGC-nosmpling). In this model, we randomly replaced the head or tail entity in the training set triples with other entities. Meanwhile, we ensured that the newly generated triples were not present in the training or test sets. These newly generated triples were used as negative samples. The ratio of positive to negative samples was also set at 1:10. The experimental results are shown in Table 3.

Table 3. Main results for FB15k-237 dataset.

Methods	MRR	H@1	H@3	H@10
SimPKGC	41.2	33.7	44.3	55.4
SimPKGC-nosampling	34.2	23.7	39.0	54.4

After introducing the negative sampling method, this study found that the improvement in the H@10 index was small, while the improvement in the H@1 index was significant. This indicates that the negative sampling function can help the model to further distinguish the target entity among the high-scoring candidate entities. At the same time, the overall performance of the model was significantly improved. In order to further study and analyze, the test set was divided into three categories: to-one, to-many and to-class, as shown in Table 4.

Table 4. Primary results for different kinds of relations in the FB15k-237 dataset.

Methods	To-One				To-Many				To-Class			
	MRR	**H@1**	**H@3**	**H@10**	**MRR**	**H@1**	**H@3**	**H@10**	**MRR**	**H@1**	**H@3**	**H@10**
SimPKGC	70.8	61.7	75.7	88.5	20.4	13.6	22.4	32.5	41.7	33.1	44.7	60.3
SimPKGC-nosampling	55.9	41.9	63.8	81.0	20.2	12.1	22.2	37.0	29.4	15.0	38.4	58.0

This study found that our negative sampling method had a significant improvement effect on the model's performance for to-one and to-class relations, especially for to-one relations, where all indicators showed significant improvement, which fully proves the rationality of the starting point of this study. For to-many relations, the improvement effect of our method was not very obvious, which this study believes is due to the high difficulty of reasoning for to-many relations.

For to-one relations, we selected a subset of relations for further analysis.

According to the data in Table 5, our proposed negative sampling method significantly improved the reasoning performance for most of the to-one relations and performed well in both frequent and non-frequent data, further proving the effectiveness of the function.

Table 5. The primary result of the FB15k-237 dataset for to-one relational data.

Serial Number	Number of Test Sets	SIMPKGC				SimPKGC-Nosampling			
		MRR	**h1**	**h3**	**h10**	**MRR**	**h1**	**h3**	**h10**
1	26	60	46.2	69.2	80.8	26.7	7.6	38.5	57.7
2	98	78.2	67.3	84.7	95.9	63.5	57.1	62.2	77.6
3	157	11.5	5.1	10.8	21.7	21.6	12.1	24.8	38.9
4	59	78.8	67.8	88.1	100	46.7	27.1	57.6	89.8
5	90	41.4	22.2	46.7	100	27.8	7.8	32.2	83.3
6	9	52.8	22.2	88.9	100	17.6	0	11.1	77.8
7	493	90.9	86.0	94.9	98.6	81.9	70.6	92.7	98.0
8	16	64.5	50.0	75.0	100	36.6	25	37.5	62.5
9	346	97.1	94.2	100	100	59.8	19.7	100	100
10	16	68.8	50.0	87.5	100	52.3	43.8	50	93.8

4.4.3. Effectiveness of Links Section

In order to eliminate the influence of extraneous text, we select a single sentence as the descriptive statement for the entity's textual description and input it into the model for training. To enhance the training performance, we adopted an entity linking approach, selecting the statement most relevant to the current triple as the entity description. In order to verify the validity of the function, we removed the link function from the original model and formed a new model (SimPKGC-nolink). The new model selected the entity's definition statement as the description, usually the first sentence. The comparative results are shown in Table 6.

Table 6. The impact of the link section on inference.

Methods	MRR	H@1	H@3	H@10
SimPKGC	41.2	33.7	44.3	55.4
SimPKGC-nolink	34.8	28.1	37.1	48.1

The comparative results demonstrate that the linking function enhances all performance metrics. It indicates that it facilitates the target entity to achieve higher scores and distinguishes it from similar and dissimilar entities. The incorporation of the linking function significantly improves the model's inference performance. It validates the effectiveness of the function and highlights the crucial role of the relevant sentences in predicting inferences.

5. Conclusions and Future Work

With the rapid advancement of pretrained language models, several PLM-based knowledge graph completion models have emerged. However, a performance gap remains between these models and state-of-the-art knowledge graph embedding models. In this study, we identified the critical factor for the suboptimal performance as the negative sampling process. To tackle this issue, we harnessed the structure information of the knowledge graph for categorization. We then adopted corresponding negative sampling methods for different categories, thereby proposing a novel PLM-based KGC model. Our experiments demonstrate the impact of negative sample collection on inference. Furthermore, the results indicate that our model achieves superior performance compared to previous models. In future work, we plan to refine the entity clustering approach and explore the potential of incorporating structural information to further enhance inference.

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

Author Contributions: Conceptualization, L.D.; methodology, L.D. and J.W.; software, J.W.; validation, J.W. and Q.S.; formal analysis, Q.S.; investigation, B.L.; resources, B.L.; data curation, J.W.; writing—original draft preparation, J.W.; writing—review and editing, L.D.; visualization, X.X.; supervision, L.D. and Q.S.; project administration, Q.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: If you need to research data, please contact the corresponding author at m21181005@nue.edu.cn.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, Vancouver, BC, Canada, 9–12 June 2008; pp. 1247–1250.
2. Suchanek, F.M.; Kasneci, G.; Weikum, G. Yago: A core of semantic knowledge. In Proceedings of the 16th International Conference on World Wide Web, Banff, AB, Canada, 8–12 May 2007; pp. 697–706.
3. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. Dbpedia: A nucleus for a web of open data. In *Proceedings of the Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Republic of Korea, 11–15 November 2007*; Proceedings, 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 722–735.
4. Dong, X.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmann, T.; Sun, S.; Zhang, W. 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, New York, NY, USA, 24–27 August 2014; pp. 601–610.
5. Sha, X.; Sun, Z.; Zhang, J. Hierarchical attentive knowledge graph embedding for personalized recommendation. *Electron. Commer. Res. Appl.* **2021**, *48*, 101071. [CrossRef]
6. Hao, Y.; Zhang, Y.; Liu, K.; He, S.; Liu, Z.; Wu, H.; Zhao, J. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Long Papers. Volume 1, pp. 221–231.
7. Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2724–2743. [CrossRef]
8. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2787–2795.
9. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014.

10. Sun, Z.; Deng, Z.-H.; Nie, J.-Y.; Tang, J. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv* **2019**, arXiv:1902.10197.
11. Balaevi, I.; Allen, C.; Hospedales, T.M. TuckER: Tensor Factorization for Knowledge Graph Completion. *arXiv* **2019**, arXiv:1901.09590.
12. Yao, L.; Mao, C.; Luo, Y. KG-BERT: BERT for knowledge graph completion. *arXiv* **2019**, arXiv:1909.03193.
13. Wang, B.; Shen, T.; Long, G.; Zhou, T.; Wang, Y.; Chang, Y. Structure-augmented text representation learning for efficient knowledge graph completion. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 1737–1748.
14. Wang, L.; Zhao, W.; Wei, Z.; Liu, J. SimKGC: Simple Contrastive Knowledge Graph Completion with Pre-trained Language Models. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; Long Papers. Volume 1, pp. 4281–4294.
15. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **2023**, *55*, 1–35. [CrossRef]
16. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex embeddings for simple link prediction. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; PMLR: Maastricht, The Netherlands, 2016; pp. 2071–2080.
17. Chao, L.; He, J.; Wang, T.; Chu, W. Pairre: Knowledge graph embeddings via paired relation vectors. *arXiv* **2020**, arXiv:2011.03798.
18. Xie, R.; Liu, Z.; Jia, J.; Luan, H.; Sun, M. Representation learning of knowledge graphs with entity descriptions. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, Arizona, 12–17 February 2016.
19. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 84–90. [CrossRef]
20. Daza, D.; Cochez, M.; Groth, P. Inductive entity representations from text via link prediction. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 798–808.
21. Wang, X.; He, Q.; Liang, J.; Xiao, Y. Language Models as Knowledge Embeddings. *arXiv* **2022**, arXiv:2206.12617.
22. Yang, J.; Jin, H.; Tang, R.; Han, X.; Feng, Q.; Jiang, H.; Yin, B.; Hu, X. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *arXiv* **2023**, arXiv:2304.13712.
23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
24. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
25. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv* **2019**, arXiv:1909.11942.
26. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
27. Sun, C.; Qiu, X.; Xu, Y.; Huang, X. How to fine-tune bert for text classification? In Proceedings of the Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, 18–20 October 2019; Proceedings 18, 2019. Springer: Berlin/Heidelberg, Germany, 2019; pp. 194–206.
28. Hakala, K.; Pyysalo, S. Biomedical Named Entity Recognition with Multilingual BERT. In Proceedings of the 5th Workshop on BioNLP Open Shared Tasks, Hong Kong, China, 4 November 2019.
29. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv* **2020**, arXiv:1910.10683.
30. Tay, Y.; Dehghani, M.; Tran, V.Q.; Garcia, X.; Wei, J.; Wang, X.; Chung, H.W.; Bahri, D.; Schuster, T.; Zheng, H.S. UL2: Unifying Language Learning Paradigms. *arXiv* **2022**, arXiv:2205.05131.
31. Zoph, B.; Bello, I.; Kumar, S.; Du, N.; Huang, Y.; Dean, J.; Shazeer, N.; Fedus, W. ST-MoE: Designing Stable and Transferable Sparse Expert Models. *arXiv* **2022**, arXiv:2202.08906.
32. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Amodei, D. Language Models are Few-Shot Learners. *arXiv* **2020**, arXiv:2005.14165.
33. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A. Training language models to follow instructions with human feedback. *arXiv* **2022**, arXiv:2203.02155.
34. Hambardzumyan, K.; Khachatrian, H.; May, J. Warp: Word-level adversarial reprogramming. *arXiv* **2021**, arXiv:2101.00121.
35. Han, X.; Zhao, W.; Ding, N.; Liu, Z.; Sun, M. Ptr: Prompt tuning with rules for text classification. *AI Open* **2022**, *3*, 182–192. [CrossRef]
36. Cui, L.; Wu, Y.; Liu, J.; Yang, S.; Zhang, Y. Template-Based Named Entity Recognition Using BART. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online, 1–6 August 2021; pp. 1835–1845.
37. Jiang, Z.; Xu, F.F.; Araki, J.; Neubig, G. How can we know what language models know? *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 423–438. [CrossRef]
38. He, F.; Liu, T.; Webb, G.I.; Tao, D. Instance-dependent pu learning by bayesian optimal relabeling. *arXiv* **2018**, arXiv:1808.02180.
39. Liu, B.; Liu, Q.; Xiao, Y. A new method for positive and unlabeled learning with privileged information. *Appl. Intell.* **2022**, *52*, 2465–2479. [CrossRef]

40. Toutanova, K.; Chen, D.; Pantel, P.; Poon, H.; Choudhury, P.; Gamon, M. Representing text for joint embedding of text and knowledge bases. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1499–1509.
41. Dettmers, T.; Minervini, P.; Stenetorp, P.; Riedel, S. Convolutional 2d knowledge graph embeddings. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
42. Wang, X.; Gao, T.; Zhu, Z.; Zhang, Z.; Liu, Z.; Li, J.; Tang, J. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Trans. Assoc. Comput. Linguist.* **2021**, *9*, 176–194. [CrossRef]