Bridge Structural Knowledge and Pre-trained Language Models for Knowledge Graph Completion

Anonymous ACL submission

Abstract

Knowledge graph completion (KGC) is a task of inferring missing triples based on existing Knowledge Graphs (KGs). Both structural and semantic information are vital for successful KGC. However, existing methods only use either the structural knowledge from the KG embeddings or the semantic information from pre-trained language models (PLMs), leading to suboptimal model performance. Moreover, since PLMs are not trained on KGs, directly using PLMs to encode triples is inappropriate. To overcome these limitations, we propose a novel model called Bridge, which jointly encodes structural and semantic information of KGs. Specifically, we strategically encode entities and relations separately by PLMs to better utilize the semantic knowledge of PLMs and enable structured representation learning via a structural learning principle. Furthermore, to bridge the gap between KGs and PLMs, we employ a self-supervised representation learning method called BYOL to fine-tune PLMs with two different views of a triple. Experiments demonstrate that Bridge outperforms the SOTA models on three benchmark datasets.

1 Introduction

Knowledge graphs (KGs) are graph-structured databases composed of triples (facts), where each triple (h, r, t) represents a relation r between a head entity h and a tail entity t. KGs such as Wikidata (Vrandečić and Krötzsch, 2014) and WordNet (Fellbaum, 2010) have a significant impact on various downstream applications such as named entity recognition (Zhou et al., 2022), relation extraction (Ren et al., 2017), and question answering (Behzad et al., 2023). Nevertheless, the effectiveness of KGs has long been hindered by the challenge of the incompleteness problem.

To address this issue, researchers have proposed a task known as Knowledge Graph Completion (KGC), which aims to predict missing relations and provides a valuable supplement to enhance KGs quality. Most existing KGC methods fall into two main categories: structure-based methods and pre-trained language model (PLMs)-based methods. Structure-based methods represent entities and relations as low-dimensional continuous embeddings, which effectively preserve their intrinsic structure (Bordes et al., 2013; Dettmers et al., 2018; Kim et al., 2022; Ge et al., 2023). While effective in KGs structure representation learning, these methods overlook the semantic knowledge associated with entities and relations. Recently, PLMs-based models have been proposed to leverage the semantic understanding captured by PLMs, adapting KGC tasks to suit the representation formats of PLMs (Yao et al., 2020; Kim et al., 2020; Wang et al., 2021a, 2022; Qiao et al., 2023).

While these models offer promising potential to enhance KGC performance, there is still space to improve: (1) Existing structure-based methods do not explore knowledge provided by PLMs. (2) Existing PLMs-based methods aim to convert KGC tasks to fit language model format and learn the relation representation from a semantic perspective using PLMs, overlooking the context of the relation in KGs. Consequently, they lack the learning of structural knowledge. For example, given a triple *(trade name, member of domain usage, metharbital)*[1], the semantic of the relation *member of domain usage* is ambiguous since "it is not a standard used term in the English[2]"; hence, PLMs may not be able to provide an accurate representation from the semantic perspective. Thus, it becomes imperative to enable the model to leverage the principle of structural learning to grasp structural knowledge and compensate for the limitations of semantic understanding. (3) Existing PLMs-based methods

[1]This is a triple from WordNet, and metharbital is an anticonvulsant drug used in the treatment of epilepsy.

[2]interpretation from ChatGPT when asking "what does *member of domain usage* mean?"

utilize PLMs directly and overlook the disparity between PLMs and triples arising from the lack of triple training during PLMs pre-training.

To address the limitations of existing methods, we propose an all-in-one framework named Bridge. To overcome the challenge of lacking structural knowledge in PLMs, we propose a structured triple knowledge learning phase. Specifically, we follow the principle that if (h, r, t) holds, then the embedding of the tail entity t should be close to the embedding of the head entity h plus the embedding of relation r, to conduct structured learning. This principle has been widely applied in traditional structured representation learning for KGs (Bordes et al., 2013; Balazevic et al., 2019), but there is no previous study that investigates this principle using PLMs-based representation. We strategically extract the embedding of h, r and t separately from PLMs, and this approach allows us to reconstruct KGs structure in the semantic embedding via the structured learning principle.

However, due to the different principles between traditional structured representation learning and PLMs, there is a gap between them since PLMs are not trained on KGs. To bridge the gap between PLMs and KGs, we fine-tune PLMs to integrate structured knowledge from KGs into PLMs. Considering the existence of one-to-many, many-to-one, and many-to-many relations in KGs (e.g. $(h_1, r, t_1), (h_1, r, t_2), (h_2, r, t_1), \cdots, (h_n, r, t_n)$ can be correct simultaneously), we opt to consider positive samples only to avoid false negatives. Therefore, we employ BYOL (Grill et al., 2020) because BYOL does not need negative samples. By taking this step, we unify the space of structural and semantic knowledge, making the integration of KGs and PLMs more reasonable.

In summary, our main contributions are:

1. We utilize structured representation learning based on a PLMs-based model to extract embeddings of entities and relations separately, which enables us to measure their spatial relations and learn structured knowledge.

2. We propose to utilize BYOL for fine-tuning PLMs to bridge the gap between structural knowledge and PLMs.

3. Experiment results on three benchmark datasets show that Bridge consistently and significantly outperforms other baseline methods.

2 Related Work

2.1 Structure-based KGC

Structure-based KGC aims to embed entities and relations into a low-dimensional continuous vector space while preserving their intrinsic structure through the design of different scoring functions. Various knowledge representation learning methods can be divided into the following categories: (1) Translation-based models, which assess the plausibility of a fact by calculating the Euclidean distance between entities and relations (Bordes et al., 2013; Ji et al., 2015; Sun et al., 2018; Ge et al., 2023); (2) Semantic matching-based models, which determine the plausibility of a fact by calculating the semantic similarity between entities and relations (Nickel et al., 2011; Yang et al., 2015; Balazevic et al., 2019; Liang et al., 2023); and (3) Neural network-based models, which employ deep neural networks to fuse the graph network structure and content information of entities and relations (Guan et al., 2018; Shang et al., 2019; Vashishth et al., 2019; Kim et al., 2022). All these structure-based models are limited to using graph structural information from KGs, and they do not leverage the rich contextual semantic information of PLMs to enrich the representation of entities and relations.

2.2 PLMs-based KGC

PLMs-based KGC refers to a method for predicting missing relations in KGs using the implicit knowledge of PLMs. KG-BERT (Yao et al., 2020) is the first work to utilize PLMs for KGC. It treats triples in KGs as textual sequences and leverages BERT (Kenton and Toutanova, 2019) to model these triples. MTL-KGC (Kim et al., 2020) utilizes a multi-task learning strategy to learn more relational properties. This strategy addresses the challenge faced by KG-BERT, where distinguishing lexically similar entities is difficult. To improve inference efficiency of KG-BERT, StAR (Wang et al., 2021a) partitions each triple into two asymmetric parts and subsequently constructs a bi-encoder to minimize the inference cost. SimKGC (Wang et al., 2022) follows the bi-encoder design of StAR and propose to utilize contrastive learning to improve the discriminative capability of the learned representation. Adopting the architecture of SimKGC, GHN (Qiao et al., 2023) develops an innovative self-information-enhanced contrastive learning approach to generate high-quality negative samples. However, all these methods simply involve fine-tuning BERT directly, disregarding both the absence of structured knowledge in BERT and the gap between BERT and KGs.

Among all the baseline models, SimKGC is the one most related to our work with two key differences. Firstly, the scoring functions in the constructive loss differ. We apply the structure-based learning principle to the scoring function, integrating structural knowledge from KGs into PLMs. In contrast, SimKGC continues to adhere to the training objective of PLMs and solely considers semantic similarity. Several studies have highlighted that modifying the scoring function is a non-trivial task and can significantly impact KGC tasks (Ji et al., 2021; Li and Yang, 2022; Ge et al., 2023). Additionally, a naive change in the scoring function overlooks the disparity between KGs and PLMs. Hence, we introduce the BYOL fine-tuning strategy to bridge the gap between KGs and PLMs.

3 Preliminary

3.1 Problem Definition

Knowledge Graph Completion The knowledge graph completion (KGC) task is to either predict the tail/head entity t given the head/tail entity h and the relation r: $(h, r, ?)$ and $(?, r, t)$, or predict relation r between two entities: $(h, ?, t)$. In this work, we focus on head and tail entity prediction.

3.2 Bootstrap Your Own Latent (BYOL)

Bootstrap Your Own Latent (BYOL) is an approach to self-supervised image representation learning without using negative samples. It employs two networks, referred to as *online* and *target* network, working collaboratively to learn from one another. The *online* network is defined by a set of weights θ, while the *target* network shares the same architecture as the *online* network but utilizes a different set of weights ξ.

Given the image x, BYOL generates two augmented views (v, v') from the image x using different augmentations. These two views (v, v') are separately processed by the *online* and the *target* encoders. The *online* network produces a representation $\mathbf{y}_\theta = f_\theta(v)$ and a projection $\mathbf{z}_\theta = g_\theta(\mathbf{y}_\theta)$, while the *target* network outputs a representation $\mathbf{y}'_\xi = f_\xi(v')$ and a projection $\mathbf{z}'_\xi = g_\xi(\mathbf{y}'_\xi)$. Next, only the *online* network applies a prediction $q_\theta(\mathbf{z}_\theta)$, creating an asymmetric between the *online* and the *target* encoders. Finally, the loss function is defined as the mean squared error between the normalized predictions and target projections :

$$\mathcal{L}_{\theta,\xi} \triangleq \|\bar{q}_\theta(\mathbf{z}_\theta) - \bar{\mathbf{z}}'_\xi\|_2^2 = 2 - 2 \cdot \frac{\langle q_\theta(\mathbf{z}_\theta), \mathbf{z}'_\xi \rangle}{\|q_\theta(\mathbf{z}_\theta)\|_2 \cdot \|\mathbf{z}'_\xi\|_2}, \tag{1}$$

where $\bar{q}_\theta(\mathbf{z}_\theta)$ and $\bar{\mathbf{z}}'_\xi$ are the $l2$-normalized term of $q_\theta(\mathbf{z}_\theta)$ and $\mathbf{z}'_\xi$.

To symmetrize the loss $\mathcal{L}_{\theta,\xi}$, BYOL swaps the two augmented views of each network, feeding v' to the *online* network and v to the *target* network to compute $\widetilde{L}_{\theta,\xi}$. During each training step, BYOL performs a stochastic optimization step to minimize $\mathcal{L}^{BYOL}_{\theta,\xi} = \mathcal{L}_{\theta,\xi} + \widetilde{L}_{\theta,\xi}$ with respect to θ only. ξ are updated after each training step using an exponential moving average of the online parameters θ as follows:

$$\xi \leftarrow \tau\xi + (1 - \tau)\theta, \tag{2}$$

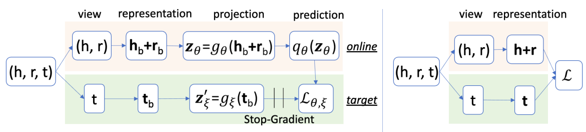
where τ is a target decay rate.

Directly predicting within the representation space can result in representations collapsing. For instance, when a representation remains constant across different views, it becomes entirely self-predictive. Therefore, the efficacy of the non-negative strategy in BYOL can be attributed to two key factors: (1) introducing a prediction network to the *online* network, establishing an asymmetry between the *online* and *target* networks, and (2) the parameters of the *target* network are updated by a slowly moving average of the *online* parameters, enabling smoother changes in the *target* representation. Both these factors work together to prevent collapsed solutions.

4 Methodology

In this section, we present Bridge structure in detail. We first introduce a structure-aware PLMs encoder, which aims to learn structure knowledge by PLMs. Then we introduce two essential modules in Bridge. The first module utilizes a fine-tuning process with BYOL to seamlessly integrate structural knowledge from KGs into PLMs, thereby bridging the gap between the two. The second module aims to learn structure-enhanced triple knowledge with PLMs. As shown in Fig.1, Bridge integrates these two modules by sequentially training two objectives.

Here, we take the tail entity prediction task $(h, r, ?)$ as an example to illustrate the procedure, and the procedure for the head entity prediction task $(?, r, t)$ will be discussed in Section 4.4.



(a) Fine-tuning PLMs with BYOL

(b) Structured Triple Knowledge Learning

Figure 1: The framework of Bridge

4.1 Structure-Aware PLMs Encoder

Existing structure-based methods do not explore leveraging PLMs, while existing PLMs-based KGC models solely rely on the semantic knowledge of PLMs. Both approaches can lead to suboptimal performance, especially when dealing with ambiguous relations. As we discussed in Section 1, the relation *member of domain usage* in the triple *(trade name, member of domain usage, metharbital)* is challenging to interpret semantically. Hence, it is essential to combine structure knowledge with semantic knowledge to achieve a structure-enhanced relation representation.

To facilitate structure representation learning, we use two BERT encoders to separately encode h, r and t. Given a triple (h, r, t), the first encoder takes the textual description of the head entity h and relation r as input, where the textual description of the head entity h is denoted as a sequence of tokens $(e_1^h, e_2^h, \cdots, e_n^h)$, and relation r is denoted as a sequence of tokens $(r_1, r_2, \cdots, r_n)$, the input sequence format is: $[CLS]\ e_1^h\ e_2^h\ \cdots\ e_n^h\ [SEP]\ r_1\ r_2\ \cdots\ r_n\ [SEP]$. The second encoder takes the textual description of the tail entity t as input, where the textual description of the tail entity t is denoted as a sequence of tokens $(e_1^t, e_2^t, \cdots, e_n^t)$, the input sequence format is: $[CLS]\ e_1^t\ e_2^t\ \cdots\ e_n^t\ [SEP]$. The design of these two encoders are illustrated in Fig.2. The embedding of h, r, t are computed by taking the mean pooling of the corresponding BERT output:

$$\mathbf{h} = MeanPooling(\mathbf{e_1^h}, \mathbf{e_2^h}, \cdots, \mathbf{e_n^h}),$$
$$\mathbf{r} = MeanPooling(\mathbf{r_1}, \mathbf{r_2}, \cdots, \mathbf{r_n}), \quad (3)$$
$$\mathbf{t} = MeanPooling(\mathbf{e_1^t}, \mathbf{e_2^t}, \cdots, \mathbf{e_n^t}).$$

To reconstruct KGs structure in the semantic embedding, we follow the widely applied principle in the KGC task that if (h, r, t) holds, then the embedding of the tail entity t should be close to the embedding of the head entity h plus the embedding of relation r. The structure scoring function

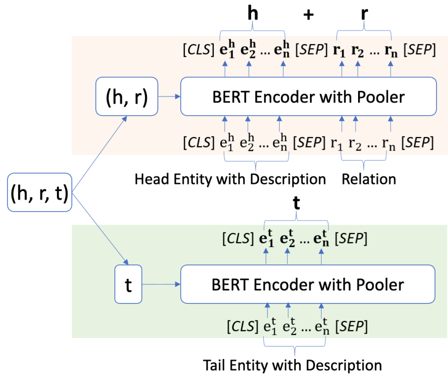


Figure 2: Structure-Aware PLMs Encoder

$\phi(h, r, t)$ of this principle is designed as follows:

$$\phi(h, r, t) = \cos(\mathbf{h} + \mathbf{r}, \mathbf{t}) = \frac{(\mathbf{h} + \mathbf{r}) \cdot \mathbf{t}}{\|\mathbf{h} + \mathbf{r}\| \|\mathbf{t}\|}. \quad (4)$$

4.2 Fine-tuning PLMs with BYOL

Previous PLM-based KGC approaches leverage PLMs directly and disregard the gap between structure knowledge and PLMs because PLMs are not trained on triples. Additionally, Bridge utilization of the traditional structure KG representation learning principle differs from that of PLMs. Therefore, strategic fine-tuning PLMs becomes necessary. Considering the existence of one-to-many, many-to-one, and many-to-many relations in KGs we exclusively consider positive samples and hence adopt BYOL (Grill et al., 2020) as it does not require negative samples. However, unlike the original BYOL model that employs two encoders to learn the representations, we leverage BYOL to initialize the parameters of encoders. This approach bridges the gap between structure and semantic knowledge, making it more feasible to integrate KGs and PLMs effectively.

As discussed in Section 3.2, BYOL generates two augmented views of the same instance, with one view serving as the input for the *online* network, and the other view as the input for the *target* network.

Here, the *online* encoder takes the textual descriptions of the head entity h and relation r as input, and produces an *online* representation $\mathbf{h_b} + \mathbf{r_b}$. The *target* encoder takes the textual descriptions of the tail entity t as input, and produces a *target* representation $\mathbf{t_b}$. The design of the encoder is elaborated in Section 4.1.

The *online* projection network g_θ takes the *online* representation $\mathbf{h_b} + \mathbf{r_b}$ as input and outputs an *online* projection representation $\mathbf{z}_\theta$:

$$\mathbf{z}_\theta = g_\theta(\mathbf{h_b} + \mathbf{r_b}) = \mathbf{W_2}[\sigma(\mathbf{W_1}[\mathbf{h_b} + \mathbf{r_b}])], \tag{5}$$

where $\mathbf{W_1}$ and $\mathbf{W_2}$ are trainable parameters, g_θ is a Multilayer Perceptron (MLP) network with one hidden layer, and $\sigma(\cdot)$ is a PReLU function.

The *target* projection network g_ξ takes the *target* representation $\mathbf{t_b}$ as input and outputs a *target* projection representation $\mathbf{z}'_\xi$:

$$\mathbf{z}'_\xi = g_\xi(\mathbf{t_b}) = \mathbf{W_4}[\sigma(\mathbf{W_3}\mathbf{t_b})], \tag{6}$$

where $\mathbf{W_3}$ and $\mathbf{W_4}$ are trainable parameters, g_ξ is a MLP network with one hidden layer, and $\sigma(\cdot)$ is a PReLU function.

The prediction network q_θ takes the *online* projection representation $\mathbf{z}_\theta$ as input and outputs a representation $q_\theta(\mathbf{z}_\theta)$ which is a prediction of the *target* projection representation $\mathbf{z}'_\xi$, the goal is to let the *online* network predict the *target* network's representation of another augmented view of the same triple:

$$q_\theta(\mathbf{z}_\theta) \approx \mathbf{z}'_\xi, \tag{7}$$

where q_θ is a MLP network with one hidden layer.

Once fine-tuning is completed, we discard the projection networks g_θ, g_ξ and the predictor network $q_\theta(\mathbf{z}_\theta)$. Only the *online* encoder and the *target* encoder are used in the subsequent module for structure triple knowledge learning.

4.3 Structured Triple Knowledge Learning

To reconstruct KGs structures in the semantic embedding, after fine-tuning PLMs with BYOL, we employ the fine-tuned *online* encoder and the *target* encoder to facilitate structure learning. The *online* BERT encoder takes the textual description of the head entity h and the relation r as input. The *target* BERT encoder takes the textual description of the tail entity t as input. Subsequently, the structure scoring function $\phi(h, r, t)$ (refer to Eq.(4)) is utilized to further train these two encoders to incorporate structure knowledge into PLMs.

This training module is indispensable because simply fine-tuning BERT using BYOL is insufficient for acquiring the adequate structure knowledge observed in training triples. We illustrate the rationality behind the training framework of each module in Section 5.4.

4.4 Head Entity Prediction

For the head entity prediction task $(?, r, t)$, we follow the principle that if (h, r, t) holds, then the embedding of the head entity h should be close to the embedding of the tail entity t minus the embedding of relation r, to conduct structure knowledge learning. Bridge separately encodes (r, t) and h using two BERT encoders. Given a triple (h, r, t), the first encoder takes the relation r and the textual description of tail entity t as input, and the input sequence format is: $[CLS]\ r_1\ r_2\ \cdots\ r_n\ [SEP]\ e_1^t\ e_2^t\ \cdots\ e_n^t\ [SEP]$. The second encoder takes the textual description of the head entity h as input, and the input sequence format is: $[CLS]\ e_1^h\ e_2^h\ \cdots\ e_n^h\ [SEP]$.

Corresponding to the Section 4.2, the *online* projection network g_θ takes the *online* representation $\mathbf{t_b} - \mathbf{r_b}$ as input and outputs an *online* projection representation $\mathbf{z}_\theta$:

$$\mathbf{z}_\theta = g_\theta(\mathbf{t_b} - \mathbf{r_b}) = \mathbf{W_6}[\sigma(\mathbf{W_5}[\mathbf{t_b} - \mathbf{r_b}])], \tag{8}$$

where $\mathbf{W_5}$ and $\mathbf{W_6}$ are trainable parameters.

The *target* projection network g_ξ takes the *target* representation $\mathbf{h_b}$ as input and outputs a *target* projection representation $\mathbf{z}'_\xi$:

$$\mathbf{z}'_\xi = g_\xi(\mathbf{h_b}) = \mathbf{W_8}[\sigma(\mathbf{W_7}\mathbf{h_b})], \tag{9}$$

where $\mathbf{W_7}$ and $\mathbf{W_8}$ are trainable parameters.

Corresponding to the Section 4.3, the structure scoring function $\phi(h, r, t)$ is designed as follows:

$$\phi(h, r, t) = \cos(\mathbf{t} - \mathbf{r}, \mathbf{h}) = \frac{(\mathbf{t} - \mathbf{r}) \cdot \mathbf{h}}{\|\mathbf{t} - \mathbf{r}\| \|\mathbf{h}\|}. \tag{10}$$

4.5 Objective and Training Process

During the Fine-tuning PLMs with BYOL phase, the loss $\mathcal{L}_{\theta,\xi}$ is calculated by Eq.(1). The *online* parameters θ are updated by a stochastic optimization step to make the predictions $q_\theta(\mathbf{z}_\theta)$ closer to $\mathbf{z}'_\xi$ for each triple, while the target parameters ϕ are updated by Eq.(2). To symmetrize this loss, we also swap the input of the *online* and *target* encoder.

During Structured Triple Knowledge Learning phase, we use contrastive loss with additive margin (Wang et al., 2022) to simultaneously optimize the structure and PLMs objectives:

$$\mathcal{L} = -log \frac{e^{(\phi(h,r,t)-\gamma)/\tau}}{e^{(\phi(h,r,t)-\gamma)/\tau} + \sum_{i=1}^{|\mathcal{N}|} e^{(\phi(h,r,t'_i)-\gamma)/\tau}}, \tag{11}$$

where τ denotes the temperature parameter, t'_i denotes the i_{th} negative tail, $\phi(h, r, t)$ is the score function as in Eq.(4) or Eq.(10), and the additive margin $\gamma > 0$ encourages the model to increase the score of the correct triple (h, r, t).

The loss $\mathcal{L}$ is computed across all positive triples in the minibatch, and entities within the same batch can serve as negatives. This extensively utilized in-batch negative strategy (Chen et al., 2020; Wang et al., 2022) enables the efficient reuse of entity embeddings for bi-encoder models.

5 Experimental Study

5.1 Datasets and Evaluation Metrics

We conduct experiments on three benchmark datasets: WN18RR (Dettmers et al., 2018), FB15k-237 (Toutanova et al., 2015), and Wikidata5M (Wang et al., 2021b). To assess the performance of Bridge and all baseline models, we employ two evaluation metrics: Hits@K and mean reciprocal rank. More details can be found in Appendix A.1.

5.2 Baseline

Structure-based methods aim to learn entity and relation embeddings by modeling relational structure in KGs. We consider the following widely used methods as baselines: TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), RotatE (Sun et al., 2018), TuckER (Balazevic et al., 2019), CompGCN (Vashishth et al., 2019), BKENE (Kim et al., 2022), CompoundE (Ge et al., 2023) and SymCL[3](Liang et al., 2023). All these methods solely rely on structural knowledge. Following the principle that the relation is a translation from the head entity to the tail entity, they design different scoring functions to measure the plausibility of a triple without leveraging the semantic knowledge of PLMs.

PLMs-based methods aim to enrich the knowledge representation by leveraging the semantic knowledge of PLMs. We consider the following PLMs-based models as baselines: KG-BERT (Yao et al., 2020), MTL-KGC (Kim et al., 2020), StAR (Wang et al., 2021a), SimKGC (Wang et al., 2022), SimKGC-SymCL[4] (Liang et al., 2023), and GHN (Qiao et al., 2023). All of these methods directly utilize semantic knowledge from PLMs, while ignoring the structural knowledge of KGs and disregarding the disparity between PLMs and KGs due to the fact that PLMs are not trained on KGs.

The information source of Structure-based models and PLMs-based methods differs. The former relies exclusively on structural knowledge derived from KGs, while the latter incorporates knowledge obtained during the pre-training process.

5.3 Overall Evaluation Results and Analysis

The performances of all models on three datasets are reported in Table 1. The experimental details can be found in the appendix A.2.

In general, with the exception of SimKGC, SimKGC-SymCL and GHN, all the other previous PLMs-based methods fall behind most structure-based methods. Meanwhile, despite the contrastive learning strategy in SimKGC, SimKGC-SymCL and GHN greatly improved performance on the WN18RR and Wikidata5M-Trans, they still lags behind structure-based methods on the FB15k-237. As claimed in Wang et al. (2022), the unsatisfactory performance on the FB15k-237 is due to the semantic ambiguity of many relations. These phenomena highlight the importance of leveraging relation context in KGs and semantic knowledge from PLMs to learn a comprehensive relation representation.

Bridge achieves superior performance compared to most of the other models. Compared with the runner-up results, the improvements obtained by Bridge in terms of MRR, Hits@3, and Hits@10 are 2.4%, 2.2%, 4.6% on WN18RR. Additionally, Hits@1 remains competitive with GHN. On the Wikidata5M-Trans dataset, Bridge exhibits substantial improvements, achieving increases of 24.7% in MRR, 26.8% in Hits@1, 25.8% in Hits@3, and 22.7% in Hits@10, respectively. On FB15k-237, Bridge achieves the best results in Hits@1 and Hits@10 while exhibiting comparable performance in Hits@3 and MRR when compared to the best results in BKENE. Considering that FB15k-237 is much denser (average degree is ~ 37 per entity) (Wang et al., 2022), BKENE likely holds an advantage in utilizing abundant neighboring information for learning entity embeddings.

Bridge outperforms state-of-the-art methods by a significant margin on Wikidata5M-Trans compared to the other two datasets. One possible reason is that Wikidata5M-Trans is larger than the other two datasets, and the abundant training data allows the fine-tuning PLMs with BYOL phase to play a more significant role, resulting in a better starting point for encoders. Further discussion is available in Section 5.4.

[3]SymCL is a plug-and-play contrastive learning approach based on relation-symmetrical structure. It can be adapt to various knowledge graph embedding methods. We report the best result achieved by SymCL on structure-based methods in Table 1.

[4]We reported the results of applying SymCL to SimKGC in Table 1.

Model	WN18RR				FB15k-237				Wikidata5M-Trans			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
Structure-based Methods												
TransE†	24.3	4.3	44.1	53.2	27.9	19.8	37.6	44.1	25.3	17.0	31.1	39.2
DistMult†	44.4	41.2	47.0	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	29.0	23.4	32.2	39.0
TuckER†	47.0	44.3	48.2	52.6	35.8	26.6	39.4	54.4	-	-	-	-
CompGCN*	47.9	44.3	49.4	54.6	35.5	26.4	39.0	53.5	-	-	-	-
BKENE*	48.4	44.5	51.2	58.4	**38.1**	29.8	**42.9**	57.0	-	-	-	-
CompoundE*	49.1	45.0	50.8	57.6	35.7	26.4	39.3	54.5	-	-	-	-
SymCL*	49.1	44.8	50.4	57.6	37.1	27.6	41.1	56.6	-	-	-	-
PLMs-based Methods												
KG-BERT*	-	-	-	52.4	-	-	-	42.0	-	-	-	-
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*	67.1	58.5	73.1	81.7	33.3	24.6	36.2	51.0	35.3	30.1	37.4	44.8
SimKGC-SymCL*	65.7	54.6	70.9	79.1	32.4	23.5	35.4	50.4	-	-	-	-
GHN*	67.8	**59.6**	71.9	82.1	33.9	25.1	36.4	51.8	36.4	31.7	38.0	45.3
Bridge	**69.4**	59.4	**74.7**	**85.9**	38.0	**31.6**	41.2	**57.4**	**45.4**	**40.2**	**47.8**	**55.6**

Table 1: Main results on WN18RR, FB15k-237 and Wikidata5M-Trans. **Bold** numbers represent the best results and underline numbers denote the runner-up results, † cites the results from Wang et al. 2022, ∗ cites the results from original papers. - indicates that the original papers do not present results related to the corresponding dataset.

Model	WN18RR				FB15k-237				Wikidata5M-Trans			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
w/o structural	58.2	45.2	64.4	79.3	31.0	24.2	31.9	44.7	30.1	27.7	30.0	38.1
w/o BYOL	70.1	59.0	72.2	80.8	38.3	30.5	40.8	56.4	40.6	33.8	40.2	50.6
SimKGC	67.1	58.5	73.1	81.7	33.3	24.6	36.2	51.0	35.3	30.1	37.4	44.8
Bridge	69.4	59.4	74.7	85.9	38.0	31.6	41.2	57.4	45.4	40.2	47.8	55.6

Table 2: Ablation study on WN18RR, FB15k-237 and Wikidata5M-Trans.

5.4 Ablation Study

To explore the effectiveness of each module, we conduct two variants of Bridge: (1) removing the structural Triple Knowledge Learning module (referred to as "w/o structural"). For inference, we use the fine-tuned *online* BERT and *target* BERT to encode $(h, r)/(r, t)$ and t/h, respectively, and rank the plausibility of each triple based on their cosine similarity (refer to Eq.(4) or Eq.(10)); (2) remove the Fine-tuning PLMs with BYOL module (referred to as "w/o BYOL"). The difference between "w/o BYOL" and SimKGC is that the former uses a structure-based scoring function, while the latter uses a semantic-based scoring function. The results are summarized in Table 2.

Effectiveness of Structured Triple Knowledge Learning: Comparing with Bridge, the results of "w/o structured" reveal that removing the Structured Triple Knowledge Learning module results in notable decreases in all metrics. This indicates that contrastive loss effectively distinguishes similar yet distinct instances. This result is consistent with the empirical studies conducted in SimKGC.

Effectiveness of Fine-tuning PLMs with BYOL: Comparing with Bridge, the results of "w/o BYOL" reveal that removing the fine-tuning BERT with BYOL module results in notable decreases across all metrics in Wikidata5M-Trans, and a minor decline in Hits@1, Hits@3, and Hits@10 on both WN18RR and FB15k-237. This phenomenon illustrates the necessity for fine-tuning PLMs. While PLMs typically utilize vast, unlabeled corpora during training to construct a comprehensive language model that embodies textual content, achieving competitive performance in particular tasks often requires an additional fine-tuning step. Meanwhile, the results also validate our previous speculation that abundant data is crucial for fine-tuning the model since Wikidata5M-Trans is larger than the other two datasets. Therefore, removing fine-tuning BERT with BYOL module has a more significant negative impact on Wikidata5M-Trans.

Compared with SimKGC, "w/o BYOL" outperforms on FB15k-237 and Wikidata5M-Trans. On WN18RR, "w/o BYOL" outperforms SimKGC in Hits@1 and MRR while being comparable in Hits@3 and Hits@10. This illustrates that our structural scoring function can effectively reconstruct KGs structures in the semantic embedding. Therefore the learned representation not only includes semantic knowledge from PLMs but also

	SimKGC		Bridge	
Triple	Rank	Top 3	Rank	Top 3
*(rio pasion, mouth of the watercourse, **Usumacinta river**)*	119	*Golfo de Paria, El Golfo de Guayaquil, Yuma River*	2	*Tabasco River, **Usumacinta river**, tzala river*
*(lewis gerhardt goldsmith, instance of, **Human**)*	11	*plant death, dispute, internet hoax*	1	***Human**, Lists of people who disappeared, Strange deaths*
*(cross country championships - short race, sport, **Athletics**)*	4	*Cross-country running, long distance race, Road run*	1	***Athletics**, Tower running, Athletics at the Commonwealth*

Table 3: Case study on the tail entity prediction $(h, r, ?)$ task using the test set of Wikidata5M-Trans. The **Bold** font represents the true tail entity. Top 3 shows the first three tail entities predicted by SimKGC and Bridge, respectively.

Triple	Rank	Top 3
*(position, hypernym, **location**)*	3	*region, space, **location***
*(take a breather, derivationally related form, **breathing time**)*	1	***breathing time**, rest, restfulness*
*(Africa, has part, **republic of cameroon**)*	14	*Eritrea, sahara, tanganyika*

Table 4: Error Analysis on the tail entity prediction $(h, r, ?)$ task on the test set of WN18RR. The **Bold** font represents the true tail entity. Top 3 shows the first three tail entities predicted by Bridge.

incorporates the context of KGs.

Furthermore, the overall computational cost of Bridge is comparable with SimKGC. More details can be found in Appendix A.3.

5.5 Case Study

We perform a case study to delve deeper into Bridge and the KGC task.

As shown in Table 3, for the first example, the top three tail entities predicted by Bridge are three rivers in Mexico and are geographically close to the true tail entity ***Usumacinta river***. However, the top three tail entities SimKGC predicted are rivers in South America. In the second example, the relation *instance of* has ambiguous semantic interpretations. SimKGC cannot accurately capture the semantics of this relation for this triple from the PLMs, resulting in incorrect predictions for the top three tail entities. Bridge can understand this relation from the structural perspective, allowing for better predictions. These two toy examples show that when the semantics of the relations are ambiguous, integrating structural knowledge can help to learn a better relation representation.

In the third example, even though Bridge accurately predicts the true tail entity ***Athletics***, the prediction *Cross-country running* made by SimKGC can be regarded as correct. *Cross-country running* and ***Athletics*** are not mutually exclusive concepts. However, the evaluation metrics consider it an incorrect answer since the triple *(cross country championships - men's short race, sport, Cross-country running)* is not present in KGs. Based on this observation, we conducted an error analysis on the WN18RR dataset to further investigate the results.

5.6 Error Analysis

Based on the above observation, we conduct an error analysis on WN18RR to further explore this phenomenon of multiple potential true tail entities.

As shown in Table 4, in the first example, Bridge ranks the true tail entity ***location*** as the third. However, the first two tail entities predicted by Bridge are correct based on human observation. In the second example, *rest* can also be a valid tail due to the fact that *rest* and ***breathing time*** are lexically similar concepts. In the third example, Bridge ranks the true tail entity ***republic of cameroon*** as 14th, attributed to the nature of the relation *has part*, which is a many-to-many relation. The first three tail entities predicted by Bridge are correct because they are all located in Africa.

Drawing from these observations, some predicted triples might be correct based on human evaluation. However, these triples might not be present in KGs. This false negative issue results in diminished performance.

To understand the impact of false negatives on the evaluation metrics, we conduct statistical analysis on the FB15k-237 dataset. More details can be found in Appendix A.4.

6 Conclusion

In this paper, we introduce Bridge, which integrates PLMs with structure-based models. Since no previous study investigates structural principle using PLMs-based representation, we jointly encode structural and semantic information of KGs to enhance knowledge representation. Further, existing work overlook the gap between KGs and PLMs due to the absence of KGs training in PLMs. To address this issue, we utilize BYOL to fine-tune PLMs. Experimental results demonstrate Bridge outperforms most baselines. Especially on Wikidata5M-Trans, the improvements in terms of MRR, Hits@1, Hits@3, and Hits@10 are 24.7%, 26.8%, 25.8%, 22.7%, respectively.

7 Limitation

Given the competitive performance of BKENE on FB15k-237, we plan to leverage graph neural networks for combining PLMs with neighboring information from KGs to fully utilize PLMs and graph neighboring knowledge.

Additionally, we intend to design more efficient evaluation metrics based on different relation properties.

8 Ethics Statement

We comply with the ACL Code of Ethics.

References

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. In *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*, pages 5184–5193. Association for Computational Linguistics.

Shabnam Behzad, Keisuke Sakaguchi, Nathan Schneider, and Amir Zeldes. 2023. Elqa: A corpus of metalinguistic questions and answers about english. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2031–2047.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26:2787–2795.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT press.

Christiane Fellbaum. 2010. Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer.

Xiou Ge, Yun Cheng Wang, Bin Wang, and C-C Jay Kuo. 2023. Compounding geometric operations for knowledge graph completion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6947–6965.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284.

Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2018. Shared embedding based neural networks for knowledge graph completion. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 247–256.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing*, volume 1, pages 687–696.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Bosung Kim, Taesuk Hong, Youngjoong Ko, and Jungyun Seo. 2020. Multi-task learning for knowledge graph completion with pre-trained language models. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1737–1743.

Jun Seon Kim, Seong Jin Ahn, and Myoung Ho Kim. 2022. Bootstrapped knowledge graph embedding based on neighbor expansion. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4123–4127.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Jiayi Li and Yujiu Yang. 2022. Star: Knowledge graph embedding by scaling, translation and rotation. In *International Conference on AI and Mobile Services*, pages 31–45. Springer.

Ke Liang, Yue Liu, Sihang Zhou, Wenxuan Tu, Yi Wen, Xihong Yang, Xiangjun Dong, and Xinwang Liu. 2023. Knowledge graph contrastive learning based on relation-symmetrical structure. *IEEE Transactions on Knowledge and Data Engineering*.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 809–816.

Zile Qiao, Wei Ye, Dingyao Yu, Tong Mo, Weiping Li, and Shikun Zhang. 2023. Improving knowledge graph completion with generative hard negative mining. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5866–5878.

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

Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3060–3067.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2018. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1499–1509.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2019. Composition-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1911.03082*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. 2021a. Structure-augmented text representation learning for efficient knowledge graph completion. In *Proceedings of the Web Conference 2021*, pages 1737–1748.

Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 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)*, pages 4281–4294.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021b. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2020. Kg-bert: Bert for knowledge graph completion.

Kang Zhou, Yuepei Li, and Qi Li. 2022. Distantly supervised named entity recognition via confidence-based multi-class positive and unlabeled learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7198–7211.

Dataset	#Ent	#Rel	#Train	#Valid	#Test
WN18RR	40,943	11	86,835	3,034	3,134
FB15k-237	14,541	237	272,115	17,535	20,466
Wikidata5M-Trans	4,594,485	822	20,614,279	5,133	5,163

Table 5: Statistics of the Datasets. Columns 2-6 represent the number of entities, relations, triples in the training set, triples in the validation set, triples in test set, respectively.

Model	# Total Training Epoch	# Total Training Time
SimKGC	8	3331s
Bridge	9	3550s

Table 6: Comparisons of model efficiency of Bridge and SimKGC on WN18RR.

A Appendix

A.1 Datasets and Evaluation Metrics

WN18RR is a subset of WordNet (Fellbaum, 1998), and FB15k-237 is a subset of Freebase (Bollacker et al., 2008). For textual descriptions of entities, we use the data from KG-BERT (Yao et al., 2020) for WN18RR and FB15k-237 datasets, and the data from SimKGC (Wang et al., 2022) for Wikidata5M-Trans dataset. The statistics are shown in Table 5.

Hits@K indicates the proportion of correct entities ranked in the top k positions, while MRR represents the mean reciprocal rank of correct entities. MRR and Hit@k are reported under the filtered setting (Bordes et al., 2013), where the filtered setting excludes the scores of all known true triples from the training, validation, and test sets. The computation of all metrics takes averaging over two directions: head entity prediction and tail entity prediction.

A.2 Bridge Setups

We use the pre-trained bert-base-uncased (English) model as the initialized encoder. In the fine-tuning PLMs with BYOL module, we conduct training on the WN18RR, FB15k-237, and Wikidata5M datasets for 2, 2, and 1 epoch(s), respectively. The seed is 0, and the initial learning rate used for these datasets are $4 * 10^{-4}, 3 * 10^{-5}, 4 * 10^{-5}$. Subsequently, in the structural triple knowledge learning module, we perform training for 7, 10, and 1 epoch(s) on the same datasets, respectively. The corresponding initial learning rates are $1*10^{-4}, 1*10^{-5}, 3*10^{-5}$. The batch size, additive margin γ of contrastive loss, and the temperature τ are consistent across all datasets, set as 1024, 0.02, and 0.05, respectively. We impose a maximum limit of 50 tokens for entity descriptions and employ AdamW optimizer (Kingma and Ba, 2015) with linear learning rate decay. Grid search is utilized to tune the optimal hyperparameters on the validation set. We employ Pytorch[5] to implement Bridge and conduct it on a server with one A100 GPU.

A.3 Efficiency of Bridge

As GHN does not provide source code, we run SimKGC [6] on WN18RR and conduct an efficiency comparison with Bridge. We employ the same convergence criteria, halting the training process when the improvement in MRR is less than 0.05. Table 6 illustrates the model efficiency of Bridge and SimKGC on WN18RR using a single A100 GPU with a batch size of 1024. In Bridge, the Fine-tuning PLMs with BYOL step converges in 2 epochs, and the Structured Triple Knowledge Learning step achieves convergence in 7 epochs (9 epochs in total). The total training time is 3550 seconds. SimKGC converges in 8 epochs, and the total training time is 3331 seconds. Consequently, the overall computational cost of Bridge is comparable with SimKGC.

A.4 Human Evaluation

The human evaluation results are shown in Table 7.

We randomly sample 100 wrong predictions based on Hits@1 for head entity prediction and tail entity prediction tasks, respectively. For the tail entity prediction task, 30% predictions are false negative, and for the head entity prediction task, 26% predictions are false negative. The majority of these false negatives are attributed to one-to-many, many-to-one, and many-to-many relations properties, whereas the Hits@1 metric assumes that all relations are one-to-one. This analysis demonstrates the underestimation of model performance by existing metrics and highlights the need for employing different metrics to address relations of varying properties. On the other hand, the proportion of “unknown" is also relatively high in both tasks, indicating the presence of noisy data within the KGs. This also presents a potential avenue for future research on enhancing KGC performance in noisy KGs.

[5]https://pytorch.org/

[6]https://github.com/intfloat/SimKGC

task	correct	wrong	unknown
$(h, r, ?)$	30%	48%	22%
$(?, r, t)$	26%	50%	24%

Table 7: Results of human evaluation on the FB15k-237 test set. The category labeled as “unknown” indicates annotators are unable to determine the correctness of the prediction.