I Know What You Do Not Know: Knowledge Graph Embedding via Co-distillation Learning

Yang Liu
State Key Laboratory for Novel Software Technology
Nanjing University, China
yliu20.nju@gmail.com

Zequn Sun
State Key Laboratory for Novel Software Technology
Nanjing University, China
zqsun.nju@gmail.com

Guangyao Li
State Key Laboratory for Novel Software Technology
Nanjing University, China
gyli.nju@gmail.com

Wei Hu*
State Key Laboratory for Novel Software Technology
National Institute of Healthcare Data Science
Nanjing University, China
whu@nju.edu.cn

ABSTRACT

Knowledge graph (KG) embedding seeks to learn vector representations for entities and relations. Conventional models reason over graph structures, but they suffer from the issues of graph incompleteness and long-tail entities. Recent studies have used pre-trained language models to learn embeddings based on the textual information of entities and relations, but they cannot take advantage of graph structures. In the paper, we show empirically that these two kinds of features are complementary for KG embedding. To this end, we propose CoLE, a **Co**-distillation **L**earning method for KG **E**mbedding that exploits the complementarity of graph structures and text information. Its graph embedding model employs Transformer to reconstruct the representation of an entity from its neighborhood subgraph. Its text embedding model uses a pre-trained language model to generate entity representations from the soft prompts of their names, descriptions, and relational neighbors. To let the two model promote each other, we propose co-distillation learning that allows them to distill selective knowledge from each other's prediction logits. In our co-distillation learning, each model serves as both a teacher and a student. Experiments on benchmark datasets demonstrate that the two models outperform their related baselines, and the ensemble method CoLE with co-distillation learning advances the state-of-the-art of KG embedding.

CCS CONCEPTS

• **Computing methodologies → Neural networks**.

KEYWORDS

knowledge graph, link prediction, co-distillation learning

CIKM '22, October 17–21, 2022, Atlanta, GA, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557355

ACM Reference Format:
Yang Liu, Zequn Sun, Guangyao Li, and Wei Hu. 2022. I Know What You Do Not Know: Knowledge Graph Embedding via Co-distillation Learning. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557355

1 INTRODUCTION

A knowledge graph (KG) is a multi-relational graph in which each node represents an entity and the directed edge has a label indicating the specific relation between two entities. An edge in KGs is a triplet in the form of (*head entity, relation, tail entity*), such as (*Kobe Bryant, profession, Athlete*). KGs play an important role in a variety of knowledge-driven applications such as question answering and recommender systems [14]. However, KGs are typically incomplete in the real world [11], which affects the performance of downstream tasks. Researchers propose the task of link prediction to predict and complete the missing edges using KG embeddings [21]. Existing KG embedding models [31] have primarily focused on exploring graph structures, including scoring the edge plausibility [2, 8, 24], reasoning over paths [12, 15, 20], as well as convolution or aggregation over neighborhood subgraphs [5, 23, 27]. We refer to these studies as structure-based models. Learning from graph structures is indifferent to what the name of an entity or relation is, but suffers from the incompleteness and sparseness issues, making it difficult to predict triplets of long-tail entities with few edges.

In recent years, knowledge probing studies such as LAMA [19] have revealed that pre-trained language models (PLMs for short, such as BERT [9]) have the ability to store some factual and commonsense knowledge obtained from large amounts of textual corpus, encouraging increased interest in probing PLMs to complete KGs. PLM-based KG embedding models convert a triplet to a natural language-style sequence by splicing the names of entities and relations, such as "*Kobe Bryant profession Athlete*". The sequence is then encoded by a PLM. The output representations are used to predict or generate the masked entity [22]. PLM-based KG embedding models do not suffer from the incompleteness issue since the PLM has been trained using external open-domain corpus.

The two lines of KG embedding techniques are currently being studied separately. But we would like to show that structure-based and PLM-based models have potential complementarity. As

[*] Wei Hu is the corresponding author.

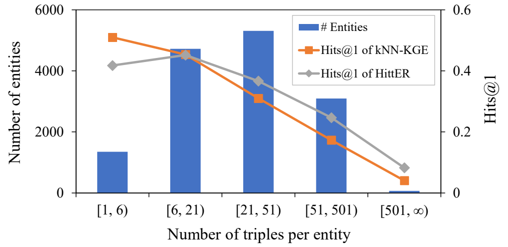


Figure 1: Histogram: entities groups of FB15K-237 based on the number of triplets per entity. Line chart: the Hits@1 performance of the PLM-based model kNN-KGE [35] and structure-based model HittER [5] in predicting the incomplete triplets of the entities in each group. 长尾节点

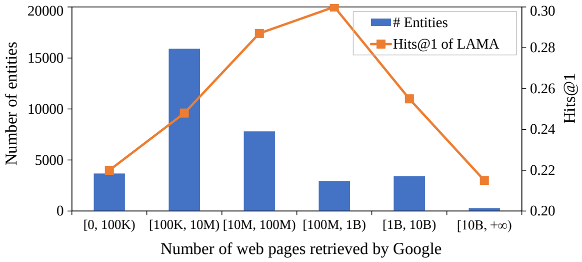


Figure 2: Histogram: entities groups of T-Rex [10] and Google-RE [19] based on the number of web pages per entity retrieved by Google using the entity name as keywords. Line chart: the Hits@1 performance of LAMA [19] in predicting the missing triplets of the entities in each group.

a preliminary study, we partition the entities in the link prediction dataset FB15K-237 [25] into several groups based on the number of edges (i.e., triplets) per entity, and evaluate the performance of the PLM-based model kNN-KGE [35] and the structure-based model HittER [5] in predicting the missing triplets of the entities in each group. Figure 1 exhibits the results. kNN-KGE performs much better than HittER in link prediction of long-tail entities (see the results on the $[1, 6)$ group), because it can benefit from the external knowledge from PLMs. HittER outperforms kNN-KGE for entities with rich edges. This indicates that they are strongly complementary. Moreover, we can see that their results decline gradually in the groups of $[6, +\infty)$. The reason for the performance decline of HittER lies in that the rich edges of an entity typically involve multi-mapping (e.g., one-to-many and many-to-one) relations, which present the widely recognized challenge for structure-based models [16, 33]. We also discover that the PLM-based model kNN-KGE may be affected by a similar issue. Entities with rich edges are usually popular and can be found in many texts. The noise, homonym, and ambiguity issues in such texts make it difficult for PLMs to recover the related content for predicting the missing triplets of a specific entity.

To investigate the aforementioned potential limitation of PLM-based models, we use the entity names in T-REx [10] and part of Google-RE [19] as query keywords to Google Search, and divide entities into several groups based on the number of retrieved web pages. Figure 2 depicts the groups along with the corresponding link prediction results of LAMA [19]. We can see that LAMA fails to perform well in link prediction of popular entities (e.g., the $[1B, 10B)$ group). We find that the PLM-based models suffer from natural language ambiguity when confronted with some popular entities with names that are similar or even the same. Another reason is that, although PLMs are trained using large amounts of textual corpus, it is difficult to retrieve the useful and to-the-point knowledge from PLMs to assist in the completion of a specific KG.

To take full advantage of the two types of models and resolve their limitations, in this paper we propose a novel approach, namely CoLE, for KG embedding via co-distillation learning between a structure-based model N-Former and a PLM-based model N-BERT. The key idea is to let the two models selectively learn from and teach each other. Specifically, CoLE consists of three components:

- The structure-based model N-Former employs Transformer [28] to reconstruct the missing entity of an incomplete triplet by leveraging the neighborhood subgraphs of the seen entity. Specifically, given an incomplete triplet $(h, r, ?)$, N-Former first reconstructs the representation of h from its neighbors. This representation is then combined with the original representation of h to further reconstruct the representation of the missing entity, t. Introducing neighborhood subgraphs in entity reconstruction can help resolve the issues of graph structure sparseness and long-tail entities.
- The PLM-based model N-BERT is built upon BERT [9] and seeks to generate the missing entity representation from a soft prompt that includes the description, neighbors, and names of the seen entity and relation. The description and neighbor information in the prompt can help retrieve the knowledge hidden in PLMs for a relevant specific entity.
- Our co-distillation learning method does not assume that one model is a teacher and the other is a student. We think that the two models are complementary in most cases. To let them benefit each other, we design two knowledge distillation (KD) objectives based on the decoupled prediction logits of the two models. The first objective seeks to transfer N-Former's knowledge concerning the high-confidence predictions into N-BERT, while the second is from N-BERT to N-Former. The prediction logits of a model are divided into two disjoint parts to calculate the two KD objectives, respectively, for selective knowledge transfer and avoiding negative transfer.

We conduct extensive experiments on benchmark datasets FB15K-237 [25] and WN18RR [8] in three setting of the structure-based, PLM-based, and ensemble link prediction. Results demonstrate that the two models, N-Former and N-BERT, achieve comparative and even better performance compared with existing related work, and the ensemble method CoLE with co-distillation learning advances the state-of-the-art of KG embedding, with a 0.294 Hits@1 score on FB15K-237 and a 0.532 Hits@1 score on WN18RR.

2 RELATED WORK

2.1 Structure-based KG Embedding

We divide structure-based KG embedding models into three groups. The first group contains the geometric models that interpret a relation in a triplet as a translation [2, 16, 33] or rotation [24] from the head entity to the tail. The second group uses bilinear functions [26] or tensor decomposition methods to score triplets [1]. The third group treats KG embedding as a deep learning task

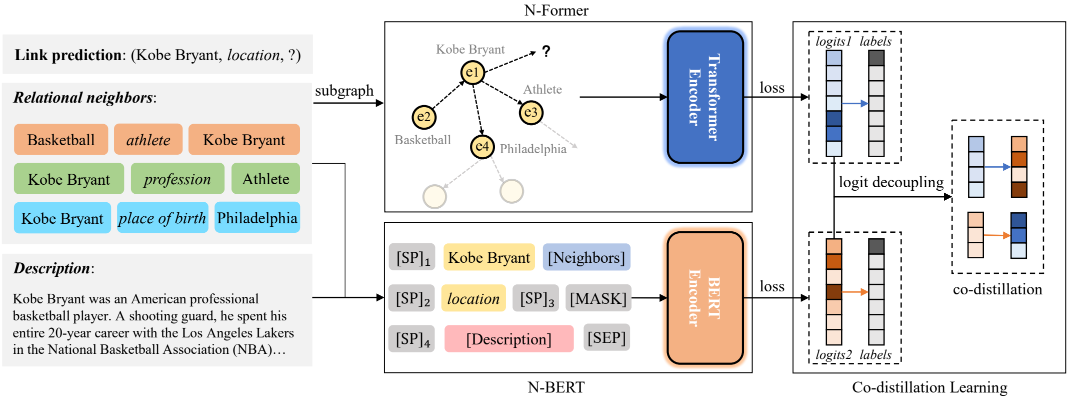


Figure 3: Framework of the proposed co-distillation learning for KG embedding. It consists of three components: the structure-based model N-Former, the PLM-based model N-BERT, and the co-distillation method to let the two models teach each other.

and explores various neural networks, including convolutional neural networks (CNNs) [8], graph convolutional networks (GCNs) [23, 27], recurrent neural networks (RNNs) [12], hyperbolic neural networks [4] and Transformers [5, 30]. Readers can refer to the surveys [21, 31] for an overview of the research progress. Our N-Former is a deep model based on Transformer [28]. CoKE [30] does not consider the neighborhood subgraphs. HittER [5] uses Transformer in a GNN manner. It has an entity Transformer to aggregate each relational neighbor of an entity, and uses a context Transformer to aggregate the neighbor representations. Despite the difference in network architectures between HittER and N-Former, their key difference lies in the different objectives. HittER uses Transformers as a neighborhood aggregator to represent an entity (like CompGCN), while N-Former is for entity reconstruction from incomplete triplets.

2.2 PLM-based KG Embedding

Unlike structure-based KG embedding models, some recent studies leverage PLMs to complete KGs by converting incomplete triplets to natural language queries. LAMA [19] is a knowledge probing model which first reveals that PLMs can capture factual knowledge present in the training data and natural language queries structured as cloze statements are able to acquire such knowledge without fine-tuning the PLMs. However, some strong restrictions, like manually constructed prompts and only predicting entities with single token names, hinder LAMA for the KG embedding task. KG-BERT [34] is the first model which applies PLMs to KG embedding, turning triplets into natural language sentences by simply concatenating entities' names and relations' names, then fine-tuning BERT for the sequence classification task. Following KG-BERT, PKGC [17] leverages manual templates to construct coherent sentences that take full advantage of PLMs. It further adds entity definitions and attributes as support information. KG-BERT and PKGC are both triplet classification models. However, using the triplet classification models for link prediction is very time-consuming. They usually assume all entities appearing in KGs are candidates, so they need numerous inference steps for one incomplete triplet, which is impractical when KGs are huge. To predict the missing entities in one inference step, MLMLM [6] adds more than one [MASK] token in the prompts to predict entities with multi-token names, while kNN-KGE [35] utilizes PLMs to learn an initial representation for each entity from its description.

Please note that our work is different from the studies injecting knowledge into PLMs to improve natural language processing (NLP) tasks. We focus on the knowledge transfer and mutual enhancement between structure- and PLM-based models. Our N-BERT is a knowledge probing model and also differs from text-enhanced embedding studies [18, 29] that integrate the text and structure features to represent entities.

3 APPROACH

This section introduces the proposed approach, CoLE.

3.1 Notations

A KG is denoted as a three-tuple $(\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ is the set of entities, $\mathcal{R}$ is the set of relations, and $\mathcal{T}$ is the set of triplets. In a KG, an entity or relation is typically represented with a Uniform Resource Identifier (URI). For example, the URI of Kobe Bryant in Freebase is `/m/01kmd4`, which, however, is not human-readable. In our approach, we assume each entity $e \in \mathcal{E}$ and relation $r \in \mathcal{R}$ to have a human-readable literal name like "Kobe Bryant", which is denoted by N_e and N_r, respectively. Our approach CoLE also leverages the textual descriptions of entities to generate prompt templates. The description of an entity e is denoted by D_e. We use $\mathbf{E}$ to denote an embedding. For example, $\mathbf{E}_e$ denotes the embedding of entity e, and $\mathbf{E}_r$ denotes the embedding of relation r.

3.2 Framework Overview

Figure 3 shows the overall framework of CoLE. The objective is to predict the missing entity in an incomplete triplet, such as (Kobe Bryant, *location*, ?) in FB15K-237. The available information used to support this prediction includes the relational neighbors of Kobe Bryant and its textual description. The proposed N-Former first takes as input the subgraph to reconstruct a representation for Kobe Bryant. The representation together with the initial embeddings of Kobe Bryant and the relation *location* are then fed into N-Former to reconstruct a representation for the missing entity denoted as a special placeholder "[MASK]" in the input. The output "[MASK]"

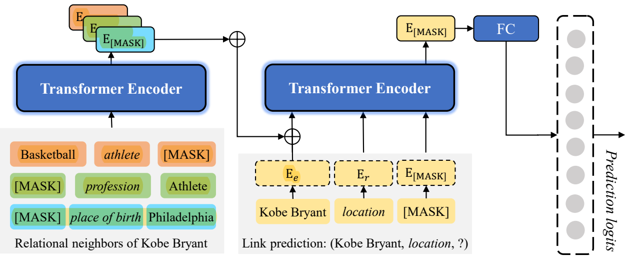


Figure 4: Architecture of the proposed N-Former.

Table 1: Prompt templates used in N-BERT.

Triplet prompt	$\texttt{prompt}_T(h,r)$ = [CLS] N_h [SEP] N_r [SEP] [MASK] [SEP] D_h [SEP]
	$\texttt{prompt}_T(r,t)$ = [CLS] [MASK] [SEP] N_r [SEP] N_t [SEP] D_t [SEP]
Relational prompt	$\texttt{prompt}_R(h,r)$ = [CLS] $[\mathrm{SP}]^r_1$ N_h $[\mathrm{SP}]^r_2$ N_r $[\mathrm{SP}]^r_3$ [MASK] $[\mathrm{SP}]^r_4$ D_h [SEP]
	$\texttt{prompt}_R(r,t)$ = [CLS] $[\mathrm{SP}]^r_1$ [MASK] $[\mathrm{SP}]^r_2$ N_r $[\mathrm{SP}]^r_3$ N_t $[\mathrm{SP}]^r_4$ D_t [SEP]
Neighbor prompt	$\texttt{prompt}_N(h,r)$ = [CLS] $[\mathrm{SP}]^r_1$ N_h [Neighbors] $[\mathrm{SP}]^r_2$ N_r $[\mathrm{SP}]^r_3$ [MASK] $[\mathrm{SP}]^r_4$ D_h [SEP]
	$\texttt{prompt}_N(r,t)$ = [CLS] $[\mathrm{SP}]^r_1$ [MASK] $[\mathrm{SP}]^r_2$ N_r $[\mathrm{SP}]^r_3$ N_t [Neighbors] $[\mathrm{SP}]^r_4$ D_t [SEP]

representation is used to compute the prediction logits. Furthermore, the PLM-based model N-BERT constructs a prompt from the descriptions, neighbors and names of entities as the input of BERT. The missing entity is also replaced with the placeholder “[MASK]”. The output representation of BERT for “[MASK]” is used to predict the missing entity. We assume that the two models are complementary, and that they should teach each other what they are good at. We propose a decoupled knowledge co-distillation method. It first divides the logits into two parts, one concerning the “high-confidence” predictions of N-Former and the other for N-BERT. Then, it enables bidirectional knowledge transfer by minimizing the KL divergence of the partial logits from the two models.

3.3 Neighborhood-aware Transformer

Figure 4 shows the architecture of the proposed N-Former with Transformer [28] as the backbone. N-Former is built on the idea of recursive entity reconstruction, which enables the model to see more for prediction while remaining robust to graph incompleteness and sparseness. We introduce the technical details below.

3.3.1 Entity Reconstruction from a Triplet. Given a triplet (h, r, t), we use the placeholder “[MASK]” to take the place of entity t, and our objective is to reconstruct the representation of t from the incomplete triplet $(h, r, [\mathrm{MASK}])$. We feed the input embeddings of $(h, r, [\mathrm{MASK}])$, which are randomly initialized, into Transformer and get the output representation for [MASK]:

$$\mathbf{E}^n_{[\mathrm{MASK}]} = \texttt{Transformer}(\mathbf{E}^0_h, \mathbf{E}^0_r, \mathbf{E}^0_{[\mathrm{MASK}]}), \tag{1}$$

where n denotes the number of self-attention layers in Transformer. With the help of self-attention, the output representation of [MASK] can capture the information from the entity h and relation r [30]. We use it as the reconstruction representation of entity t, i.e., $\mathbf{E}_t = \mathbf{E}^n_{[\mathrm{MASK}]}$, and define the prediction logits P_t as

$$\mathbf{P}_t = \texttt{softmax}\big(\mathbf{E}_{\mathrm{ENT}} \cdot \texttt{MLP}(\mathbf{E}_t)\big), \tag{2}$$

where $\mathbf{E}_{\mathrm{ENT}}$ is the embedding matrix for all entities, and $\texttt{MLP}()$ denotes a multi-layer perceptron for representation transformation. We can then calculate the cross-entropy loss between the prediction logits $\mathbf{P}_t$ and the corresponding labels $\mathbf{L}_t$ as follows:

$$\mathcal{L}_{\mathrm{triplet}}(t) = \texttt{CrossEntropy}(\mathbf{P}_t, \mathbf{L}_t), \tag{3}$$

where $\texttt{CrossEntropy}(\mathbf{x}, \mathbf{y}) = -\sum_j \mathbf{x}_j \log \mathbf{y}_j$, which computes the cross-entropy loss between two vectors.

3.3.2 Entity Reconstruction from Neighborhood. The entity reconstruction from a triplet described above can be used in a recursive manner to consider the neighborhood subgraphs. For example, for the incomplete triplet $(h, r, [\mathrm{MASK}])$, in addition to the prediction loss in Eq. (2) that only considers the input embeddings of h and r, we think that the embedding of h can also benefit from its representation reconstructed from its relational neighbors, which also lets the model see more to help with the prediction of long-tail entities. Let $\texttt{Neighbor}(h) = \{(h', r') \,|\, (h', r', h) \in \mathcal{T}\}$ denote the set of relational neighbors of entity h. Please note that we add a reverse triplet (t, r^-, h) for each triplet (h, r, t) in the KG, such that we only need to consider the incoming edges of an entity, i.e., the triplets whose tails are this entity. We use each relational neighbor of h to reconstruct a representation and sum up all these representations as the embedding of h, denoted by $\mathbf{E}^S_{Neighbor}$:

$$\mathbf{E}^S_{Neighbor} = \textstyle\sum_{(h',r') \in \texttt{Neighbor}(h)} \texttt{Transformer}(\mathbf{E}^0_{h'}, \mathbf{E}^0_{r'}, \mathbf{E}^0_{[\mathrm{MASK}]}), \tag{4}$$

where the Transformer is the same one as in Eq. (1). To let the neighborhood information of h be used in link prediction, we further expand Eq. (1) by combining the input embedding of h $\mathbf{E}^0_h$ with the embedding reconstructed from relational neighbors $\mathbf{E}^S_{Neighbor}$:

$$\mathbf{E}^S_{[\mathrm{MASK}]} = \texttt{Transformer}(\texttt{mean}(\mathbf{E}^0_h, \mathbf{E}^S_{Neighbor}), \mathbf{E}^0_r, \mathbf{E}^0_{[\mathrm{MASK}]}), \tag{5}$$

where $\texttt{mean}()$ returns the average of vectors. Given the output representation, we can compute the prediction logits $\mathbf{P}^S_t$ in the same way as Eq. (2). The overall loss of N-Former is defined as

$$\mathcal{L}_{\mathrm{structure}} = \sum_{(h,r,t) \in \mathcal{T}} \big(\mathcal{L}_{\mathrm{triplet}}(t) + \texttt{CrossEntropy}(\mathbf{P}^S_t, \mathbf{L}_t)\big). \tag{6}$$

3.4 Neighborhood-aware BERT

It has been widely acknowledged that PLMs like BERT [9] capture some structured knowledge [19], which can be leveraged for link prediction in KGs. PLMs are trained with a large amount of open-domain corpora, but link prediction is about a specific entity in a KG. The key challenge for PLM-based link prediction lies in how to retrieve the relevant knowledge from PLMs. Given an incomplete triplet $(h, r, [\mathrm{MASK}])$, a typical solution is to introduce the textual description of the entity for constructing a sentence-like prompt [35] as *triplet prompt* shown in Table 1, where [CLS], [SEP] are the special tokens used to separate the different parts of the prompt, and the missing entity is replaced with a placeholder [MASK]. It is worth noting that we do not introduce inverse relations to convert an incomplete triplet $([\mathrm{MASK}], r, t)$ as $(t, r^-, [\mathrm{MASK}])$. This is because it is a non-trivial task to get the name of a reverse relation r^- given the relation r. For an incomplete triplet $(h, r, [\mathrm{MASK}])$, its prompt sentence is denoted by $\texttt{prompt}(h, r)$. For an incomplete triplet $([\mathrm{MASK}], r, t)$, its prompt sentence is $\texttt{prompt}(r, t)$. The prompt sentences would be fed into a PLM (BERT, in our work) to get the representation of [MASK]:

$$\mathbf{E}_{[\mathrm{MASK}]} = \texttt{BERT}(\texttt{prompt}_T(h, r)), \tag{7}$$

and we use it as the representation of t.

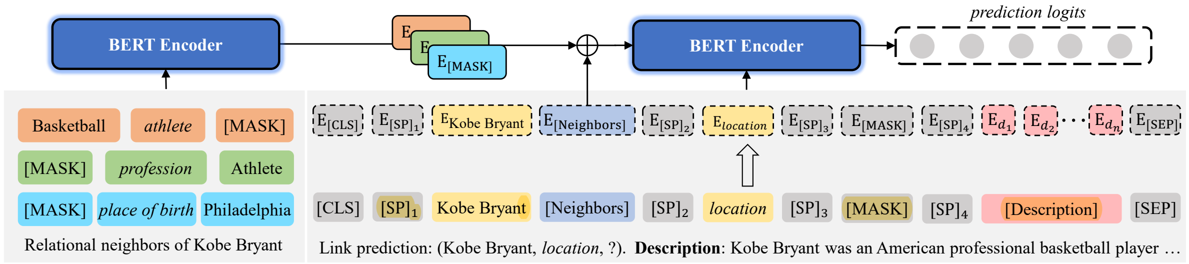


Figure 5: Architecture of the proposed N-BERT.

To get the predicted entities in one inference step, we need to add the names of entities as new tokens into the vocabularies of PLMs. But the random initial embeddings for these new tokens perform poorly, as they do not own any external knowledge from open-domain corpora. Following [35], we use the embeddings learned from the description prompts as initial representations for entities:

$$\mathbf{E}_{\mathrm{h}}^{init} = \mathtt{BERT}(\text{“}\textit{The description of } [\mathrm{MASK}] \textit{ is } D_h.\text{”}), \tag{8}$$

where D_h is the textual description of the entity h. To make our prompts more expressive and take full advantage of the PLMs, we introduce two types of additional textual information in prompts.

3.4.1 Soft Prompts. Given that PLMs are designed for NLP, it is convincing that more coherent prompts would utilize the knowledge in PLMs better, while our prompts concatenated by [SEP] are obviously unexpressive. Some works [17, 19] design manual templates to make the prompts more coherent, which is impractical for KGs with numerous relations. On the contrary, we utilize some adjustable soft prompts to make our prompt more expressive. Inspired by PKGC [17], we replace the special [SEP] tokens between other tokens with relation-aware soft prompts to get a new kind of prompts, which is named *relational prompt* in Table 1, where $[\mathrm{SP}]_i^r (i = 1, 2, 3, 4)$ denotes the i-th soft prompt for the relation r and i indicates the position where the corresponding soft prompt is to be inserted. In implementation, $[\mathrm{SP}]_i^r$ is a special token added into the vocabulary and related to the relation r, and its representation is randomly initialized.

3.4.2 Neighborhood Prompts. Recent studies [6, 17, 35] have shown that more support textual information leads to more promising performance, and the descriptions and attributes of entities have been widely used. However, the relational knowledge in KGs has not been explored for prompts, which may potentially facilitate the PLMs for link prediction. To take full advantage of the neighboring triplets in the KGs, we introduce the contextual embeddings of entities as additional support information, which can also be regraded as entity-aware soft prompts. Let $\mathtt{InNeighbor}(h) = \{(h', r') \mid (h', r', h) \in \mathcal{T}\}$ and $\mathtt{OutNeighbor}(h) = \{(r', t') \mid (h, r', t') \in \mathcal{T}\}$ denote the sets of relational neighbors of entity h. Similar to Section 3.3.2, we use neighbors of h to reconstruct a representation and sum up all these representations as the embedding of h, denoted by $\mathbf{E}_{Neighbor}^{T}$:

$$\begin{aligned}\mathbf{E}_{Neighbor}^{T} = &\sum_{(h',r')\in\mathtt{InNeighbor}(h)} \mathtt{BERT}(\mathtt{prompt}_R(h', r')) \\ &+ \sum_{(r',t')\in\mathtt{OutNeighbor}(h)} \mathtt{BERT}(\mathtt{prompt}_R(r', t')),\end{aligned} \tag{9}$$

where the BERT is the same one as in Eq. (7). After adding the neighbor information, our neighborhood prompt is named *neighbor prompt* in Table 1, where [Neighbor] $= \mathbf{E}_{Neighbor}^{T}$ is a representation vector. After adding these two types information, we can obtain the representations of entities to be predicted as

$$\begin{aligned}\mathbf{E}_h^T &= \mathtt{BERT}(\mathtt{prompt}_N(r, t)), \\ \mathbf{E}_t^T &= \mathtt{BERT}(\mathtt{prompt}_N(h, r)),\end{aligned} \tag{10}$$

and we can get the prediction logits $\mathbf{P}_h^T$ and $\mathbf{P}_t^T$ in the same way as Eq. (2). The overall loss of N-BERT is defined as follows:

$$\mathcal{L}_{\text{text}} = \textstyle\sum_{(h,r,t)\in\mathcal{T}} \big(\mathtt{CrossEntropy}(\mathbf{P}_h^T, \mathbf{L}_h) + \mathtt{CrossEntropy}(\mathbf{P}_t^T, \mathbf{L}_t)\big), \tag{11}$$

where $\mathbf{P}_h$ and $\mathbf{L}_h$ are the prediction logits and labels for predicting the head entity, $\mathbf{P}_t$ and $\mathbf{L}_t$ are the prediction logits and labels for predicting the tail entity, respectively, given the triple (h, r, t).

3.5 Co-distillation Learning

Considering the complementarity between N-Former and N-BERT, we propose to transfer the knowledge mutually by knowledge distillation, which is known as co-distillation. The loss for traditional knowledge distillation with a fixed teacher model and a student model to be optimized is a combination of the classification loss and the Kullback Leibler (KL) divergence loss, defined as follows:

$$\mathcal{L}_s = \mathtt{CrossEntropy}(\mathbf{P}_s, \mathbf{L}) + \mathtt{KL}(\mathbf{P}_t \,||\, \mathbf{P}_s), \tag{12}$$

where $\mathcal{L}_s$ denotes the knowledge distillation loss for the student model. $\mathbf{P}_t$ and $\mathbf{P}_s$ denote the probabilities of all classes predicted by the teacher model and the student model, respectively. $\mathbf{L}$ denotes the label vectors for a given instance.

For our N-Former and N-BERT, we empirically find that the scores for target entities both become higher in the training process. It is reasonable because both of our models are trained to predict the target entities as top-1. However, the scores for non-target entities are varying obviously. Given that N-Former is a structure-based model while N-BERT is a PLM-based model, we think that the scores for non-target entities are also important to quantize the knowledge of models. Therefore, the key point for co-distillation is to transfer the predicted probabilities of non-target entities mutually, rather than only to deliver the probabilities of target entities.

Recently, the importance of non-target classes has drawn more attention. DKD [36] decouples the classical knowledge distillation loss into two parts to release the potential of knowledge contained in non-target classes. Motivated by this, we also emphasize the importance of non-target entities for KG embedding. However, unlike the scenario for classical knowledge distillation with a fixed teacher, the knowledge is not guaranteed for co-distillation because a model can be the teacher and the student at the same time. Hence, we designed a heuristic method for selective knowledge transfer.

In the rest of this section, we take the task of predicting tail entities as an example. For an incomplete triplet $(h, r, [\text{MASK}])$, we obtain the prediction logits $\mathbf{P}_t^S$ and $\mathbf{P}_t^T$ from N-Former and N-BERT, respectively. When N-Former is the teacher model, we rank all entities in descending order according to $\mathbf{P}_t^S$ and select half entities with higher scores. Then, we select the logits of these entities from $\mathbf{P}_t^S$ and $\mathbf{P}_t^T$ which are denoted by $\mathbf{P}_t^{S_1}$ and $\mathbf{P}_t^{T_1}$, respectively. The decoupled loss for $\mathbf{P}_t^{S_1}$ and $\mathbf{P}_t^{T_1}$ is

$$\mathcal{L}_{KD}(\mathbf{P}_t^{S_1}, \mathbf{P}_t^{T_1}) = \mathsf{KL}(\mathbf{b}_t^{S_1} \,||\, \mathbf{b}_t^{T_1}) + \mathsf{KL}(\hat{\mathbf{P}}_t^{S_1} \,||\, \hat{\mathbf{P}}_t^{T_1}), \tag{13}$$

where $\mathbf{b}_t^{S_1} = [p_1, 1 - p_1] \in \mathbb{R}^{1\times 2}, \mathbf{b}_t^{T_1} = [p_2, 1 - p_2] \in \mathbb{R}^{1\times 2}$ denote the binary probabilities in terms of target entities, assuming that p_1 and p_2 are the probabilities for target entities from N-Former and N-BERT, respectively. $\hat{P}_t^{S_1}$ and $\hat{P}_t^{T_1}$ denote the probabilities excluded target entities from $P_t^{S_1}$ and $P_t^{T_1}$, respectively. When N-BERT is the teacher model, we can also get $\mathbf{P}_t^{S_2}$ and $\mathbf{P}_t^{T_2}$ in the same way.

3.6 Put It All Together

The proposed approach CoLE uses co-distillation learning to interact with N-Former and N-BERT for bidirectional knowledge transfer. The learning losses for N-Former and N-BERT are

$$\begin{aligned}\mathcal{L}_{\text{N-BERT}} &= \alpha\, \mathcal{L}_{KD}(\mathbf{P}_t^{S_1}, \mathbf{P}_t^{T_1}) + (1-\alpha)\, \mathtt{CrossEntropy}(\mathbf{P}_t^{T}, \mathbf{L}_t),\\ \mathcal{L}_{\text{N-Former}} &= \beta\, \mathcal{L}_{KD}(\mathbf{P}_t^{T_2}, \mathbf{P}_t^{S_2}) + (1-\beta)\, \mathtt{CrossEntropy}(\mathbf{P}_t^{S}, \mathbf{L}_t),\end{aligned} \tag{14}$$

where $\mathbf{L}_t$ denotes the label vectors for predicting t. α and β are hyper-parameters for balance. N-Former and N-BERT are optimized jointly. For each mini-batch, losses are computed from the same data and the two models are updated depending on their losses, respectively. The training process is presented in Algorithm 1. In the inference process, given an incomplete triplet, we combine the prediction probabilities of N-Former and N-BERT as the final output probabilities by weighted averaging to rank candidates.

4 EXPERIMENTS

In this section, we report the experimental results of the proposed approach CoLE. The source code is available from GitHub.[1]

4.1 Experiment Setup

4.1.1 Datasets. We use two benchmark datasets FB15K-237 [25] and WN18RR [8] to train and evaluate our approach. The two datasets remove some inverse edges from their previous versions (i.e., FB15K and WN18 [2]) to prevent the leakage of test triplets into the training process.

4.1.2 Evaluation Protocol. For each triplet (h, r, t) in the test set, we obtain two incomplete triplets, $(h, r, [\text{MASK}])$ and $(t, r^-, [\text{MASK}])$, for tail entity prediction and head entity prediction, respectively. r^- denotes the reverse relation for r. The queries are fed into N-Former to get output representations. Then the prediction logits are calculated by an inner product between the output representations and all entity embeddings. We finally obtain the ranking lists for candidate entities by sorting the logits in descending order. For N-BERT or CoLE, the ranking lists are obtained in the same way. We employ Hits@k ($k = 1, 3, 10$) and MRR under the filtered setting [2] to assess the performance. Averaged results on head entity prediction and tail entity prediction are reported.

Algorithm 1: Training process of CoLE.

```
Input: Training triplet set $\mathcal{T}_{\text{train}}$, validation triplet set $\mathcal{T}_{\text{valid}}$,
       test triplet set $\mathcal{T}_{\text{test}}$, entity names and descriptions.
Output: Candidate ranking list for each incomplete triplet.
Initialize model parameters and input embeddings;
Generate masked triplets and prompts from $\mathcal{T}_{\text{train}}$;
for $epoch \leftarrow 1$ to $max\_epoch\_num$ do
    for $step \leftarrow 1$ to $max\_step\_num$ do
        $b \leftarrow$ sample a training batch;
        for $(h, r, [MASK])$ in $b$ do
            Compute the logits and loss Eq. (6) of N-Former;
            Compute the logits and loss Eq. (11) of N-BERT;
            Compute the losses Eq. (14) of CoLE;
        Get the overall loss and update models;
    Validate CoLE using $\mathcal{T}_{\text{valid}}$;
    if early stop then break;
Test CoLE using $\mathcal{T}_{\text{test}}$;
```

Table 2: Selected hyper-parameter values for CoLE.

	Dim.	# Layers	# Heads	α	β
FB15K-237	256	8	2	0.5	0.8
WN18RR	256	12	4	0.5	0.7

4.1.3 Implementation Details. We implement our method with PyTorch and all experiments are conducted on a workstation with two Intel Xeon Gold 6326 CPUs, 512GB memory and a NVIDIA RTX A6000 GPU. We leverage the BERT-base model[2] as the PLM for N-BERT, and a vanilla Transformer encoder [28] for N-Former. We employ the AdamW optimizer and a cosine decay scheduler with linear warm-up for optimization. We determine the hyper-parameter values by using the grid search based on the MRR performance on the validation set. We select the layer number of Transformer in $\{1, 2, 4, 8, 12, 16\}$, the head number in $\{1, 2, 4, 8\}$, the batch size in $\{1024, 2048, 4096, 8192\}$, the learning rate for N-Former in $\{$1e-4, 2e-4, 3e-4, 5e-4, 1e-3$\}$, the learning rate for N-BERT in $\{$1e-5, 3e-5, 5e-5, 1e-4$\}$, the learning rate for co-distillation in $\{$1e-5, 3e-5, 5e-5, 1e-4$\}$, the values of α and β in $\{0.1, 0.2, \ldots, 0.9\}$. Table 2 lists the picked values for important hyper-parameters.

4.2 Baselines

- Structure-based models. For structure-based KG embedding, we compare N-Former against 11 representative link prediction models. We choose three geometric models, including TransE [2], RotatE [24] and DualE [3]. We also choose the latest tensor decomposition model TuckER [1]. The left baselines are all based on deep neural networks. ConvE [8] and

[1] https://github.com/nju-websoft/CoLE

[2] Downloaded from https://huggingface.co/.

Table 3: Link prediction results compared with structure-based baselines.

Model	FB15K-237				WN18RR			
	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR
TransE	0.231	0.367	0.528	0.329	0.013	0.400	0.528	0.223
ConvE	0.237	0.356	0.501	0.325	0.400	0.440	0.520	0.430
RotatE	0.241	0.375	0.533	0.338	0.428	0.492	0.571	0.476
TuckER	0.266	0.394	0.544	0.358	0.443	0.482	0.526	0.470
CoKE	0.272	0.400	0.549	0.364	0.450	0.496	0.553	0.484
CompGCN	0.264	0.390	0.535	0.355	0.443	0.494	0.546	0.479
ATTH	0.252	0.384	0.540	0.348	0.443	0.499	0.573	0.486
DualE	0.237	0.363	0.518	0.330	0.440	0.500	0.561	0.482
ConEx	0.271	0.403	0.555	0.366	0.448	0.493	0.550	0.481
M^{2}GCN	0.275	0.398	**0.565**	0.362	0.444	0.498	0.572	0.485
HittER	**0.279**	0.409	0.558	**0.373**	**0.462**	**0.516**	**0.584**	**0.503**
N-Former	0.277	**0.412**	0.556	0.372	0.443	0.501	0.578	0.486
N-Former$_{\text{co-distilled}}$	**0.279**	**0.412**	0.556	**0.373**	0.446	0.504	0.581	0.489

Table 4: Link prediction results compared with PLM-based baselines.

Model	FB15K-237				WN18RR			
	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR
KG-BERT	-	-	0.420	-	0.041	0.302	0.524	0.216
MLMLM	0.187	0.282	0.403	0.259	0.440	0.542	0.611	0.502
kNN-KGE	0.280	0.404	0.550	0.370	0.525	0.604	0.683	0.579
N-BERT	0.287	0.420	0.562	0.381	0.529	0.607	0.686	0.583
N-BERT$_{\text{co-distilled}}$	**0.293**	**0.426**	**0.570**	**0.387**	**0.532**	**0.607**	**0.689**	**0.585**

ConEx [7] use convolutions to capture the interactions between entities and relations. CompGCN [27] and M^{2}GCN [32] leverage GCNs. ATTH [4] is a hyperbolic embedding model to capture the hierarchical patterns in a KG. We also compare N-Former with CoKE [30] and HittER [5], which are the most relevant studies based on the Transformer architecture. CoKE [30] only focuses on modeling a single triplet, while HittER [5] introduces entity neighborhood as the contextual information but without entity reconstruction.

- PLM-based models. For PLM-based KG embedding, we pick 3 representative models as baselines, including KG-BERT [34], MLMLM [6] and kNN-KGE [35]. KG-BERT is a triplet classification model and only utilizes entity names and relation names as text information. It can also do link prediction but is time-consuming. MLMLM and kNN-KGE further utilize the descriptions of entities as additional information, which are both link prediction models. MLMLM adds multiple [MASK] tokens to predict entities with multi-token names, and kNN-KGE learns initial representations for all entities from descriptions before training.
- Ensemble methods. To our knowledge, no previous work considers knowledge transfer between structured-based and PLM-based link prediction. We design two ensemble methods, namely ProbsMax and ProbsAvg, in CoLE. ProbsMax selects the maximum probabilities of N-Former and N-BERT for a given entity to rank. ProbAvg averages the prediction probabilities. The two variants of CoLE are denoted by ProbsMax$_{\text{co-distilled}}$ and ProbsAvg$_{\text{co-distilled}}$, respectively.

The results of all baselines are taken from their original papers, with the exception of TransE, which appeared earlier than the two datasets. We reproduce its results using OpenKE [13].

4.3 Results and Analyses

Here we report and analyze the main results.

Results of structure-based models. Experimental results for structure-based models are shown in Table 3. It takes 7.9 hours and 6.2 hours to train N-Former with the best parameters on FB15k-237 and WN18RR, respectively. We can observe that N-Former achieves competitive performance on the two datasets. On FB15K-237, the performance of N-Former is nearly close to that of the state-of-the-art model HittER. Both models are based on the Transformer architecture and leverage neighborhood information. They outperform CoKE stably which neglects such context. On WN18RR, our model does not perform best. It is slightly inferior to HittER and comparable with other competitive methods. We think this is because WN18RR has more hierarchical structures and HittER benefits from its hierarchical architecture to capture such patterns. The characteristics of the dataset are also beneficial for ATTH and DualE, despite their underwhelming performance on FB15K-237. ATTH leverages several hyperbolic operations to distinguish the hierarchies. DualE can capture complicated relations with unified operations, including translation and rotation. We also notice that N-Former gets further improvement after co-distillation. It further indicates the complementarity between N-Former and N-BERT, and co-distillation makes them benefit from each other. We give more analyses shortly in Section 4.5.

Table 5: Link prediction results of ensemble methods.

Model	FB15K-237				WN18RR			
	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR
N-Former	0.277	0.412	0.556	0.372	0.443	0.501	0.578	0.486
N-BERT	0.287	0.420	0.562	0.381	0.529	0.607	0.686	0.583
ProbsMax	0.291	0.427	0.570	0.386	0.517	0.600	0.685	0.574
ProbsAvg	**0.294**	**0.430**	**0.574**	**0.389**	<u>0.530</u>	**0.609**	<u>0.692</u>	<u>0.585</u>
N-Former$_{\text{co-distilled}}$	0.279	0.412	0.556	0.373	0.446	0.504	0.581	0.489
N-BERT$_{\text{co-distilled}}$	<u>0.293</u>	0.426	0.570	<u>0.387</u>	**0.532**	0.607	0.689	<u>0.585</u>
ProbsMax$_{\text{co-distilled}}$	0.292	<u>0.428</u>	<u>0.571</u>	<u>0.387</u>	0.518	0.595	0.684	0.574
ProbsAvg$_{\text{co-distilled}}$	**0.294**	**0.430**	**0.574**	**0.389**	**0.532**	<u>0.608</u>	**0.694**	**0.587**

Table 6: Ablation results of N-Former and N-BERT.

Model	FB15K-237				WN18RR			
	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR
N-Former$_{\text{w/o neighbor}}$	0.269	0.399	0.548	0.362	0.437	0.478	0.523	0.466
N-Former	0.277	0.412	0.556	0.372	0.443	0.501	0.578	0.486
N-BERT$_{\text{w/o soft prompt}}$	0.264	0.389	0.527	0.354	0.510	0.579	0.672	0.563
N-BERT$_{\text{w/o description}}$	0.271	0.408	0.556	0.368	0.414	0.488	0.571	0.467
N-BERT$_{\text{w/o neighbor}}$	0.283	0.418	0.561	0.378	0.514	0.598	0.685	0.573
N-BERT	0.287	0.420	0.562	0.381	0.529	0.607	0.688	0.582

Results of PLM-based models. The results for PLM-based models are shown in Table 4. It takes 8.8 hours and 3.2 hours to train N-BERT with the best parameters on FB15k-237 and WN18RR, respectively. We can see that N-BERT even without distillation outperforms other baselines on both datasets. Among PLM-based models, KG-BERT does not work well since its prompt sentences do not give enough evidence, showing that additional information is important for PLM-based models. kNN-KGE and MLMLM use the descriptions of entities for link prediction and they obtain a significant improvement. However, the extent of improvement differs due to their utilization of the descriptions of entities. In addition to adding the descriptions as support information into the prompt sentences, kNN-KGE further learns new representations of entities based on the descriptions before the model training. Thanks to our soft prompts and neighbor information in prompt sentences, N-BERT outperforms kNN-KGE stably. We also notice that N-BERT obtains an improvement with co-distillation, similar to N-Former.

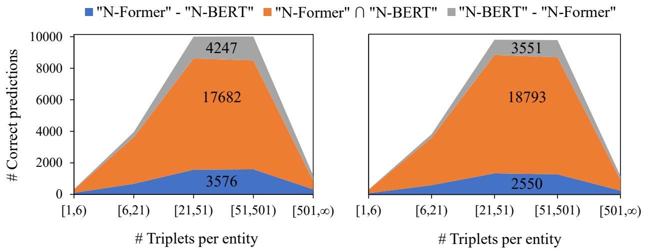
Results of ensemble methods. Table 5 lists the results of ensemble methods. It takes 8.7 hours to co-distill N-Former and N-BERT on FB15k-237 and 1.4 hours on WN18RR, with the best parameters. We notice that both ensemble methods achieve a further improvement no matter whether we leverage co-distillation. The improvement is in accord with the observation found by existing work [7] that ensemble learning with a simple combination can increase the performance. For example, the baseline ensemble methods ProbsMax and ProbsAvg can improve both N-Former and N-BERT on FB15K-237. ProbsAvg performs better than ProbsMax. The maximum probabilities emphasize some wrong predictions of the two models, making ProbsMax unstable. Furthermore, the performance of our model variants is slightly better than the ensemble baselines without co-distillation. This is in line with our intuition that ensemble learning performs better with stronger base models. As co-distillation mainly transfers mutual knowledge without generating much new knowledge, such improvement is not significant. We leave it as future work to explore other ensemble methods.

4.4 Ablation Study

The results are shown in Table 6.

Ablation study for N-Former. The only additional information that we utilize for N-Former is neighboring triplets, so we conduct an ablation study to verify the influence of neighbor information. The neighbor information significantly improves N-Former on both datasets in terms of Hits@1, 0.269 → 0.277 on FB15K-237 and 0.437 → 0.443 on WN18RR. This further indicates the effectiveness of the entity neighborhood in prediction, and our proposed N-Former can capture such contextual information well.

Ablation study for N-BERT. We apply three kinds of support textual information to construct the prompts for N-BERT, namely *soft prompt*, *entity description* and *neighbor information*. We hereby conduct an ablation study on the support information to verify their effectiveness. As we can see, all of them contribute a lot to N-BERT, and the contributions are varied for different datasets. We argue that the effectiveness of different kinds of textual information is related to the graph structure of KGs. FB15k-237 is a much more dense KG with fewer entities and more relations. Soft prompts can better enhance the expression of relations between entities in this dataset. For WN18RR, entity descriptions play a more important role. This is because that WN18RR has a large number of entities (about 41K). The additional description information can work more obviously to distinguish these entities. As for the neighbor information, the Hits@1 improvement is more significant on WN18RR than that on FB15K-237 (0.514 → 0.529 vs 0.283 → 0.287). Given that



(a) Link prediction w/o distillation

(b) Link prediction with co-distillation

Figure 6: Correct predictions of N-Former and N-BERT when using no-distillation and co-distillation on FB15K-237. The blue area denotes the right triplets predicted by N-Former which exclude those that can also be predicted by N-BERT. The orange area denotes the overlap triplets predicted correctly by N-Former and N-BERT. The gray area denotes the right triplets predicted by N-BERT which exclude those that can also be predicted by N-Former.

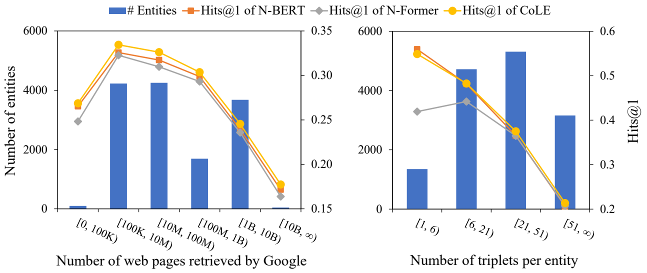


Figure 7: Hits@1 results on FB15K-237 along with the numbers of retrieved web pages and edges per entity, resp.

we only sample a few neighboring triplets, neighbor information contributes more for a more sparse KG. Overall, we demonstrate the effectiveness of all the proposed support information.

4.5 Further Analyses

Complementarity in our model. We check the correct entity prediction results of N-Former and N-BERT in overlap on FB15K-237. We also compare the results with and without co-distillation, as shown in Figure 6. When using no-distillation, 3576 triplets can be predicted correctly only by N-Former. For N-BERT, there are 4247 correct triplets. The blue and gray areas represent the two kinds of triplets. These triplets reflect the complementarity of N-Former and N-BERT. We can observe that the two areas get shrunk (i.e., the complementarity becomes weaker) if we leverage the co-distillation. In turn, the orange area, which denotes the overlap of the predicted triplets, expands. It indicates that co-distillation indeed leverages the complementarity and transfers the unique knowledge of each model mutually, thus benefiting them.

Result analyses for long-tail and popular entities. Following the practices in Section 1, we partition the entities of FB15K-237 into several groups based on the number of retrieved web pages and the number of edges per entity, respectively. When partitioning by the number of web pages, N-Former and N-BERT have the same tendency. Their performance both increases at first and then decreases along with the growing number of web pages. Moreover, CoLE outperforms them all the time. As the number of web pages cannot fully indicate whether an entity is popular or long-tail, the complementarity between N-Former and N-BERT is not obvious. We further partition entities according to the number of triplets. N-BERT surpasses N-Former by a large margin for long-tail entities, while N-Former is slightly better for popular entities. Although the scores of N-BERT and N-Former on popular entities are close, the complementarity cannot be omitted because the number of triplets related to popular entities is large. There are still many triplets that can be correctly predicted only by N-Former. Overall, the analyses above verify the complementarity between N-Former and N-BERT on both popular and long-tail entities.

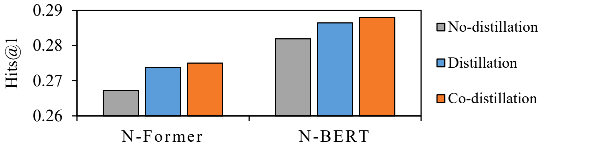


Figure 8: Hits@1 results of our models when using no-distillation, the conventional (unidirectional) distillation and our co-distillation on FB15K-237.

Comparison of different distillation methods. We also explore the effectiveness of different distillation methods. We choose the conventional (unidirectional) distillation for comparison. In this method, we fix the teacher model and only train the student model through the distillation. Figure 8 shows the results of N-Former and N-BERT without leveraging neighborhood information on FB15K-237. We observe that both methods can obtain an improvement, as distillation can generally introduce new knowledge for models. Moreover, co-distillation outperforms unidirectional distillation stably. We believe this is because that the co-distillation adopts a mutual learning strategy and allows both models to learn from each other, while the unidirectional distillation neglects such interactions. This further verifies the superiority of our co-distillation.

5 CONCLUSION AND FUTURE WORK

In this paper, we present a co-distillation learning method that seeks effective knowledge transfer and mutual enhancement between structure-based and PLM-based KG embedding models. For structure-based KG embedding, we propose N-Former that reconstructs and predicts the missing entity of an incomplete triplet based on its relational neighbors. For PLM-based KG embedding, we propose N-BERT that generates the missing entity representation by probing BERT with a prompt of entity names, descriptions, and neighbors. Our co-distillation learning method CoLE first decouples the prediction logits of the two models and then lets them teach their useful knowledge to each other by bidirectional knowledge transfer with logit distillation. Experiments on FB15K-237 and WN18RR show that N-Former and N-BERT achieve competitive and even the best results compared with existing work. The ensemble method CoLE advances the state-of-the-art of KG embedding.

In future work, we plan to investigate knowledge transfer between multi-source KGs and experiment with additional KG embedding tasks such as entity alignment.

Acknowledgments. This work was supported by National Natural Science Foundation of China (No. 61872172) and Beijing Academy of Artificial Intelligence (BAAI).

REFERENCES

[1] Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. 2019. TuckER: Tensor Factorization for Knowledge Graph Completion. In *EMNLP-IJCNLP*. ACL, Hong Kong, China, 5184–5193.

[2] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NIPS*. Curran Associates, Inc., Lake Tahoe, NV, USA, 2787–2795.

[3] Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. 2021. Dual Quaternion Knowledge Graph Embeddings. In *AAAI*. AAAI Press, online, 6894–6902.

[4] Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In *ACL*. ACL, online, 6901–6914.

[5] Sanxing Chen, Xiaodong Liu, Jianfeng Gao, Jian Jiao, Ruofei Zhang, and Yangfeng Ji. 2021. HittER: Hierarchical Transformers for Knowledge Graph Embeddings. In *EMNLP*. ACL, online, 10395–10407.

[6] Louis Clouâtre, Philippe Trempe, Amal Zouaq, and Sarath Chandar. 2021. MLMLM: Link Prediction with Mean Likelihood Masked Language Model. In *Findings of ACL*. ACL, online, 4321–4331.

[7] Caglar Demir and Axel-Cyrille Ngonga Ngomo. 2021. Convolutional Complex Knowledge Graph Embeddings. In *ESWC*. Springer, Hersonissos, Greece, 409–424.

[8] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *AAAI*. AAAI Press, New Orleans, Louisiana, USA, 1811–1818.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*. ACL, Minneapolis, MN, USA, 4171–4186.

[10] Hady ElSahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon S. Hare, Frédérique Laforest, and Elena Simperl. 2018. T-REx: A Large Scale Alignment of Natural Language with Knowledge Base Triples. In *LREC*. ELRA, Miyazaki, Japan, 3448–3452.

[11] Luis Galárraga, Simon Razniewski, Antoine Amarilli, and Fabian M. Suchanek. 2017. Predicting Completeness in Knowledge Bases. In *WSDM*. ACM, Cambridge, UK, 375–383.

[12] Lingbing Guo, Zequn Sun, and Wei Hu. 2019. Learning to Exploit Long-term Relational Dependencies in Knowledge Graphs. In *ICML*. PMLR, Long Beach, CA, USA, 2505–2514.

[13] Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. OpenKE: An Open Toolkit for Knowledge Embedding. In *EMNLP System Demonstrations*. ACL, Brussels, Belgium, 139–144.

[14] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Transactions on Neural Networks and Learning Systems* 33, 2 (2022), 494–514.

[15] Yankai Lin, Zhiyuan Liu, Huan-Bo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015. Modeling Relation Paths for Representation Learning of Knowledge Bases. In *EMNLP*. ACL, Lisbon, Portugal, 705–714.

[16] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *AAAI*. AAAI Press, Austin, Texas, USA, 2181–2187.

[17] Xin Lv, Yankai Lin, Yixin Cao, Lei Hou, Juanzi Li, Zhiyuan Liu, Peng Li, and Jie Zhou. 2022. Do Pre-trained Models Benefit Knowledge Graph Completion? A Reliable Evaluation and a Reasonable Approach. In *Findings of ACL*. ACL, Dublin, Ireland, 3570–3581.

[18] Rahul Nadkarni, David Wadden, Iz Beltagy, Noah A. Smith, Hannaneh Hajishirzi, and Tom Hope. 2021. Scientific Language Models for Biomedical Knowledge Base Completion: An Empirical Study. In *AKBC*. OpenReview.net, London, UK.

[19] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language Models as Knowledge Bases?. In *EMNLP-IJCNLP*. ACL, Hong Kong, China, 2463–2473.

[20] Petar Ristoski and Heiko Paulheim. 2016. RDF2Vec: RDF Graph Embeddings for Data Mining. In *ISWC*. Springer, Kobe, Japan, 498–514.

[21] Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. 2021. Knowledge Graph Embedding for Link Prediction: A Comparative Analysis. *ACM Transactions on Knowledge Discovery from Data* 15, 2 (2021), 14:1–14:49.

[22] Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. 2022. Sequence-to-Sequence Knowledge Graph Completion and Question Answering. In *ACL*. ACL, Dublin, Ireland, 2814–2828.

[23] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *ESWC*. Springer, Heraklion, Crete, Greece, 593–607.

[24] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *ICLR*. OpenReview.net, New Orleans, LA, USA, 1–18.

[25] 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 *EMNLP*. ACL, Lisbon, Portugal, 1499–1509.

[26] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *ICML*. PMLR, New York City, NY, USA, 2071–2080.

[27] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. Composition-based Multi-Relational Graph Convolutional Networks. In *ICLR*. OpenReview.net, Addis Ababa, Ethiopia, 1–16.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*. Curran Associates, Inc., Long Beach, CA, USA, 5998–6008.

[29] Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. 2021. Structure-Augmented Text Representation Learning for Efficient Knowledge Graph Completion. In *WWW*. ACM / IW3C2, Ljubljana, Slovenia, 1737–1748.

[30] Quan Wang, Pingping Huang, Haifeng Wang, Songtai Dai, Wenbin Jiang, Jing Liu, Yajuan Lyu, Yong Zhu, and Hua Wu. 2019. CoKE: Contextualized Knowledge Graph Embedding. *CoRR* abs/1911.02168 (2019), 1–10.

[31] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2724–2743.

[32] Shen Wang, Xiaokai Wei, Cícero Nogueira dos Santos, Zhiguo Wang, Ramesh Nallapati, Andrew O. Arnold, Bing Xiang, Philip S. Yu, and Isabel F. Cruz. 2021. Mixed-Curvature Multi-Relational Graph Neural Network for Knowledge Graph Completion. In *WWW*. ACM, Ljubljana, Slovenia, 1761–1771.

[33] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *AAAI*. AAAI Press, Québec City, Québec, Canada, 1112–1119.

[34] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: BERT for Knowledge Graph Completion. *CoRR* abs/1909.03193 (2019), 1–8.

[35] Ningyu Zhang, Xin Xie, Xiang Chen, Shumin Deng, Chuanqi Tan, Fei Huang, Xu Cheng, and Huajun Chen. 2022. Reasoning Through Memorization: Nearest Neighbor Knowledge Graph Embeddings. *CoRR* abs/2201.05575 (2022), 1–9.

[36] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. 2022. Decoupled Knowledge Distillation. *CoRR* abs/2203.08679 (2022), 1–12.