I Know What You Do Not Know:

Knowledge Graph Embedding via Co-distillation Learning

Presenter: Zhanke Zhou

2022. 10. 19

About the paper

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

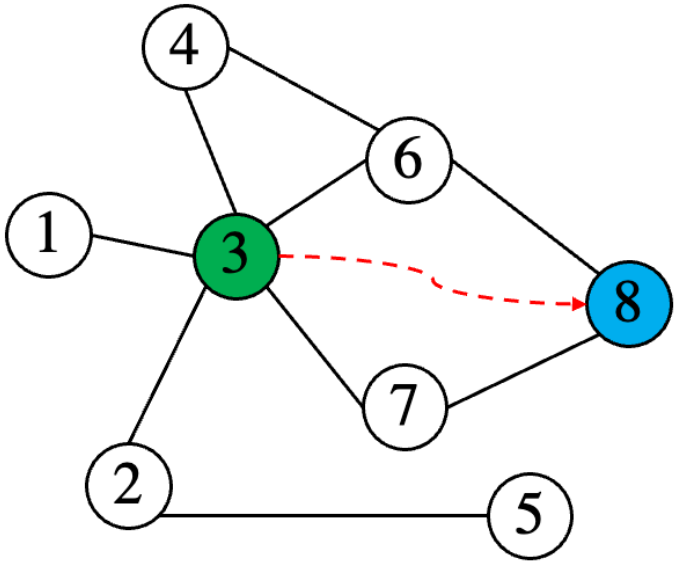
- Authors: Yang Liu, Zequn Sun, Guangyao Li, Wei Hu
- Conference: CIKM 2022
- Affiliation: Nanjing University
- Paper: https://arxiv.org/pdf/2208.09828.pdf
- Code: https://github.com/nju-websoft/CoLE

Outline

- Background
- The proposed method
- Experiment
- Summary and discussion

Background | link prediction on KG

- Link prediction task in knowledge graph
  - give a query $(e_q, r_q, ?)$, to find the target entity e_a
  - based on observed (known) edges, to predict the latent (unknown) edges



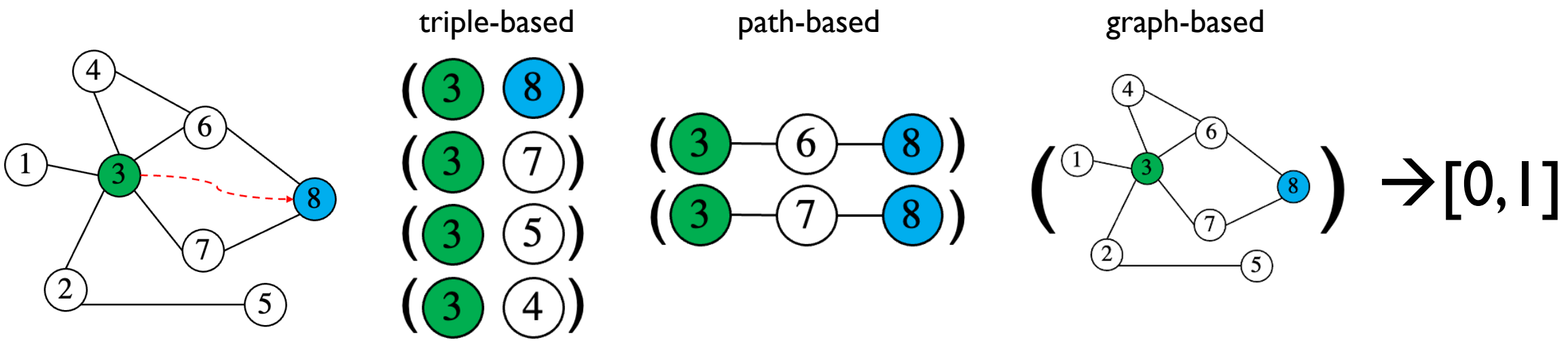
KG

$$\text{query} = (e_q, r_q, ?)$$

$$e_q \leftarrow ③, r_q \leftarrow r_1, e_a \leftarrow ⑧$$

Background | existing KG learning models

- **(SLMs) Structure learning models**: learn from the KG structure
  - triple-based (implicit structure learning)
    - $p(\mathrm{e_q}, \mathrm{r_q}, \mathrm{e_a})$ is measured by a scoring function, utilizing the representations $\mathbf{e}_\mathrm{q}, \mathbf{r}_\mathrm{q}, \mathbf{e}_\mathrm{a}$
  - path-based (explicit structure learning)
    - learn the sequential order of structures by leveraging the relational paths between $\mathrm{e_q}$ and $\mathrm{e_a}$
  - graph-based (explicit structure learning)
    - directly use the (sub-)graph structure for reasoning, capturing more complex semantics

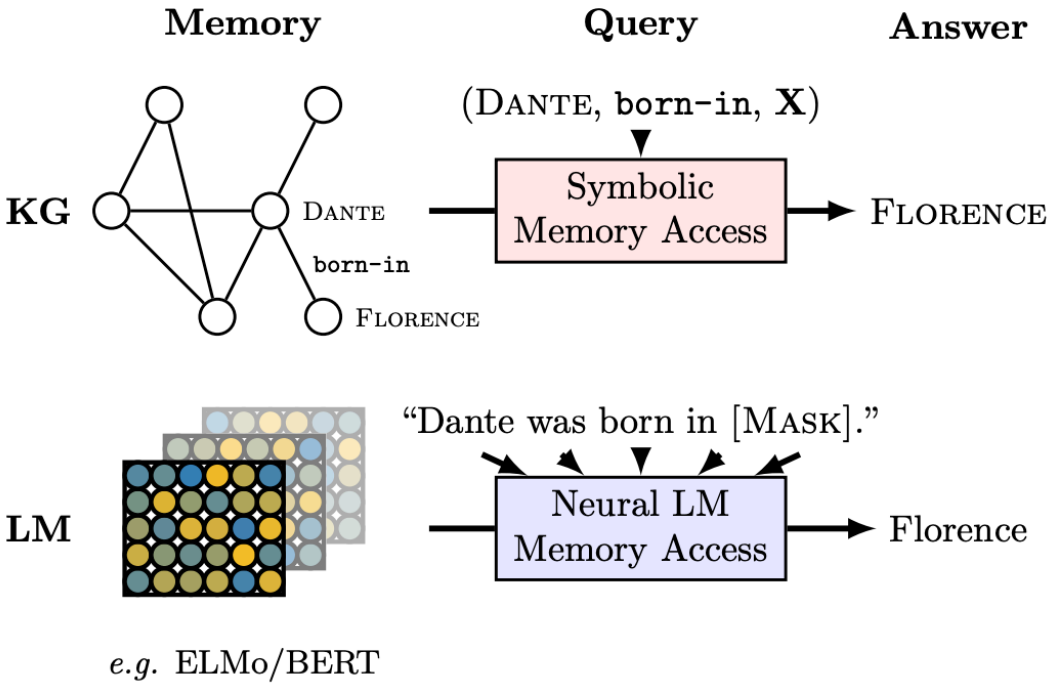


Background | using PLMs as "soft" knowledge bases

(PLMs) Pretrained language models for the relational knowledge

Usage
- asking PLMs to fill in masked tokens in sequences
- e.g., "Dante was born in [Mask]"
- building such sequences is essential for query

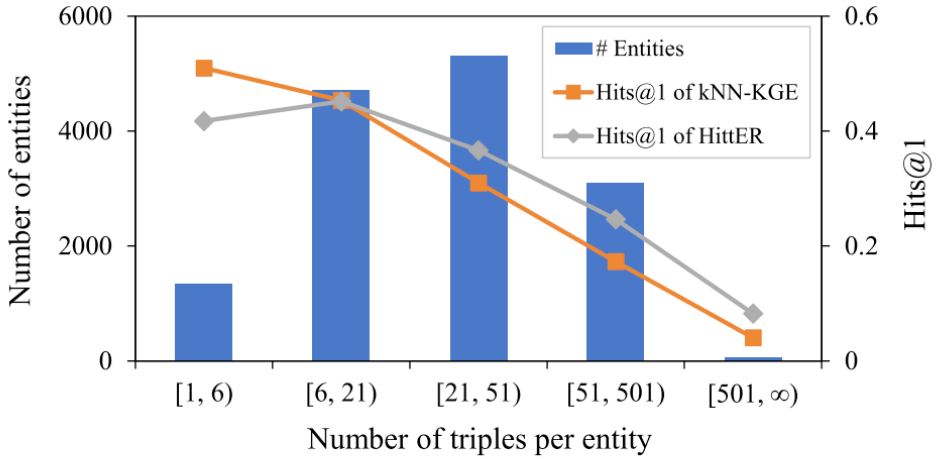
Advantages
- support an open set of queries
- can be applied in the inductive setting

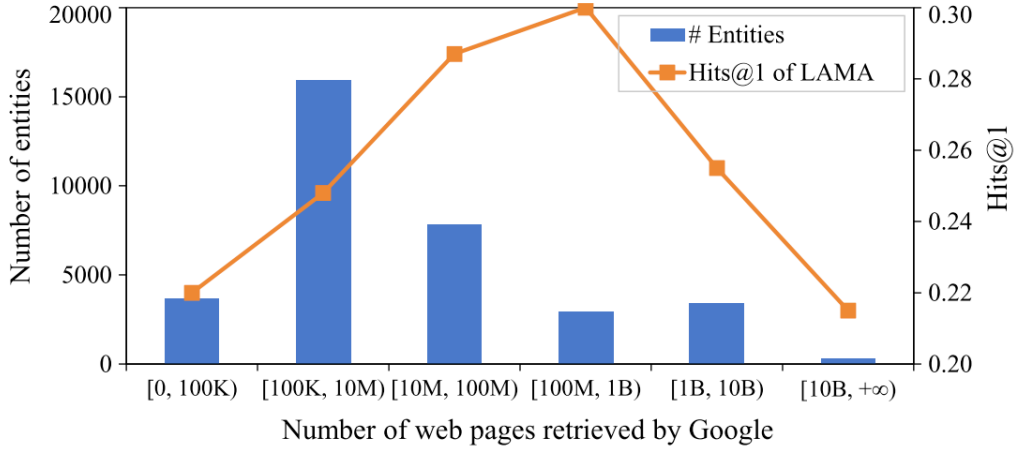


Language Models as Knowledge Bases? EMNLP 2019.

Background | motivation

SLMs (grey line) HittER uses Transformer in a GNN manner
PLMs (orange lines) KNN-KGE/LAMA that are based on BERT





Observations

For long-tail entities, PLM is better than SLM

- because it can benefit from the external knowledge

For entities with rich edges, SLM is better than PLM

- these popular entities can be found in many texts
- The noise, homonym, and ambiguity issues in such texts hinder PLM

Background | motivation

Two lines of KG learning models
(1) Structure-based models
(2) Pretrained language model

These two lines are studied separately.

So, can they be complementary?
can they help each other and boost performances?

➔ Yes, we can achieve it by knowledge distillation

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^2GCN	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
$\text{N-Former}_{\text{co-distilled}}$	**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
$\text{N-BERT}_{\text{co-distilled}}$	**0.293**	**0.426**	**0.570**	**0.387**	**0.532**	**0.607**	**0.689**	**0.585**

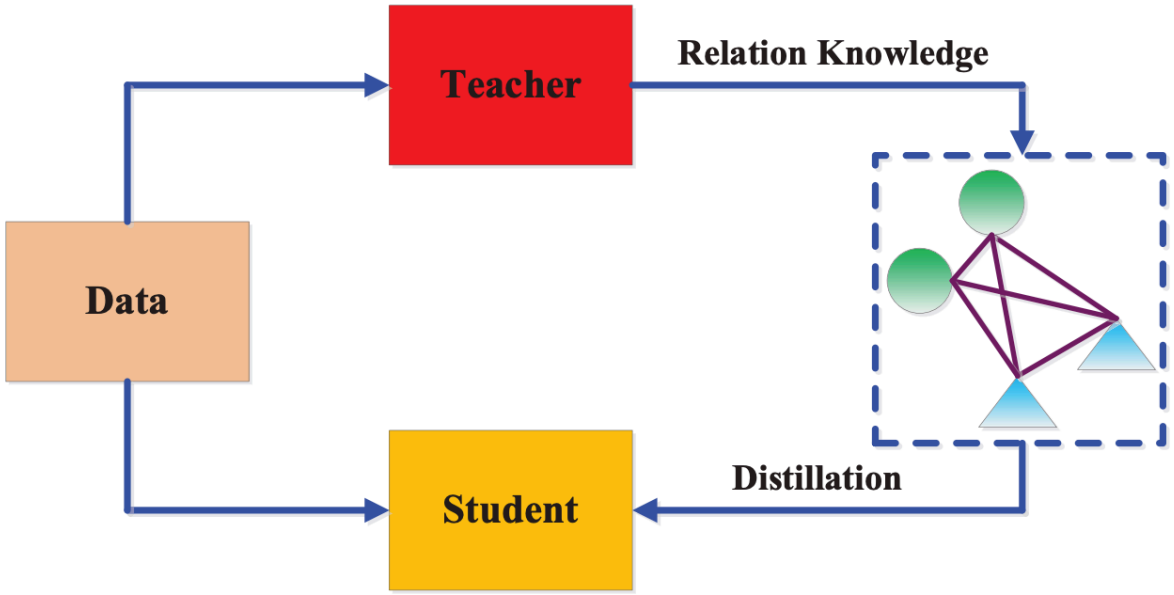
Table 5: Link prediction results of ensemble methods.

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**	0.530	**0.609**	0.692	0.585

Background | motivation

Knowledge distillation

- in short, knowledge distillation is the process of transferring knowledge from teacher model to student model



loss function for the student model

$$\mathcal{L}_s = \underbrace{\texttt{CrossEntropy}(\mathbf{P}_s, \mathbf{L})}_{\text{classification loss}} + \underbrace{\texttt{KL}(\mathbf{P}_t \,||\, \mathbf{P}_s)}_{\text{behave like the teacher}}$$

Outline

- Background
- The proposed method
- Experiment
- Summary and discussion

Method | framework

The objective is to predict the missing entity in an incomplete triplet

- such as (Kobe Bryant, location, ?) in FB15K-237

Three major components:

- one SLM, one PLM, and one distillation module

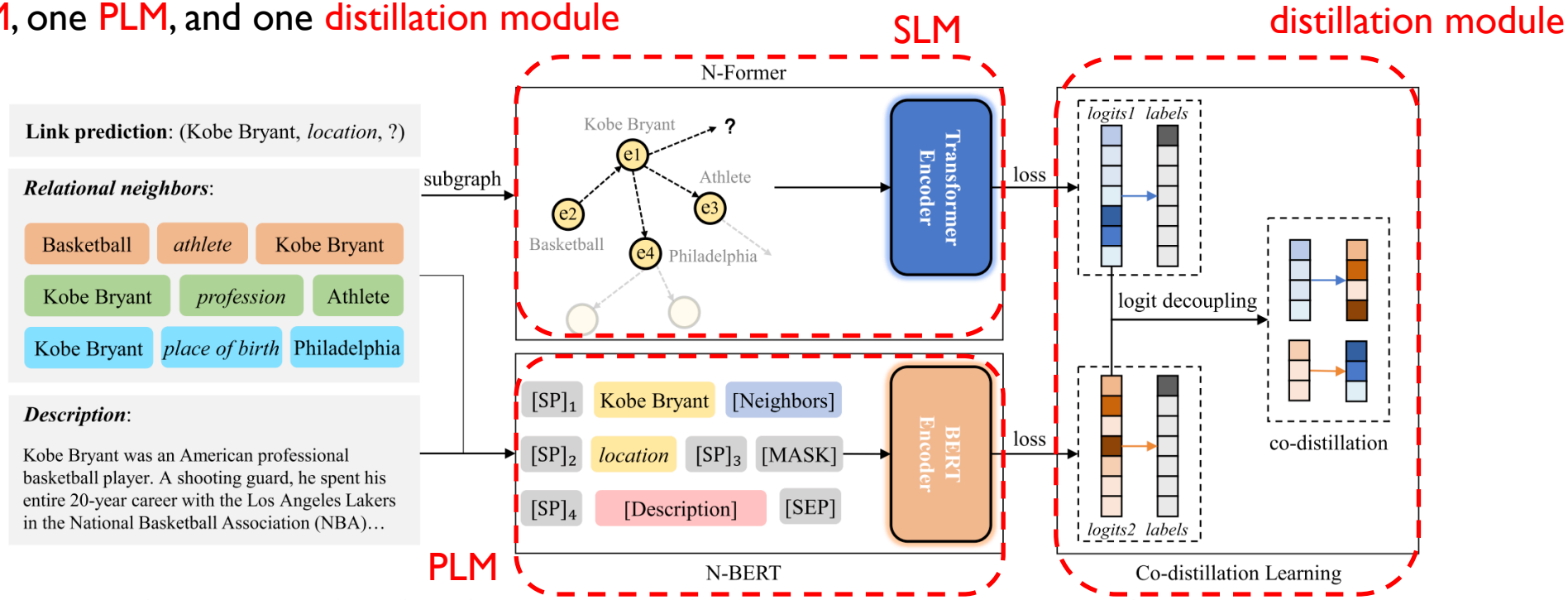


Figure 3: Framework of the proposed co-distillation learning for KG embedding. It consists of three components: the structure-based model N-Former, the PLM-based model N-BERT, and the co-distillation method to let the two models teach each other.

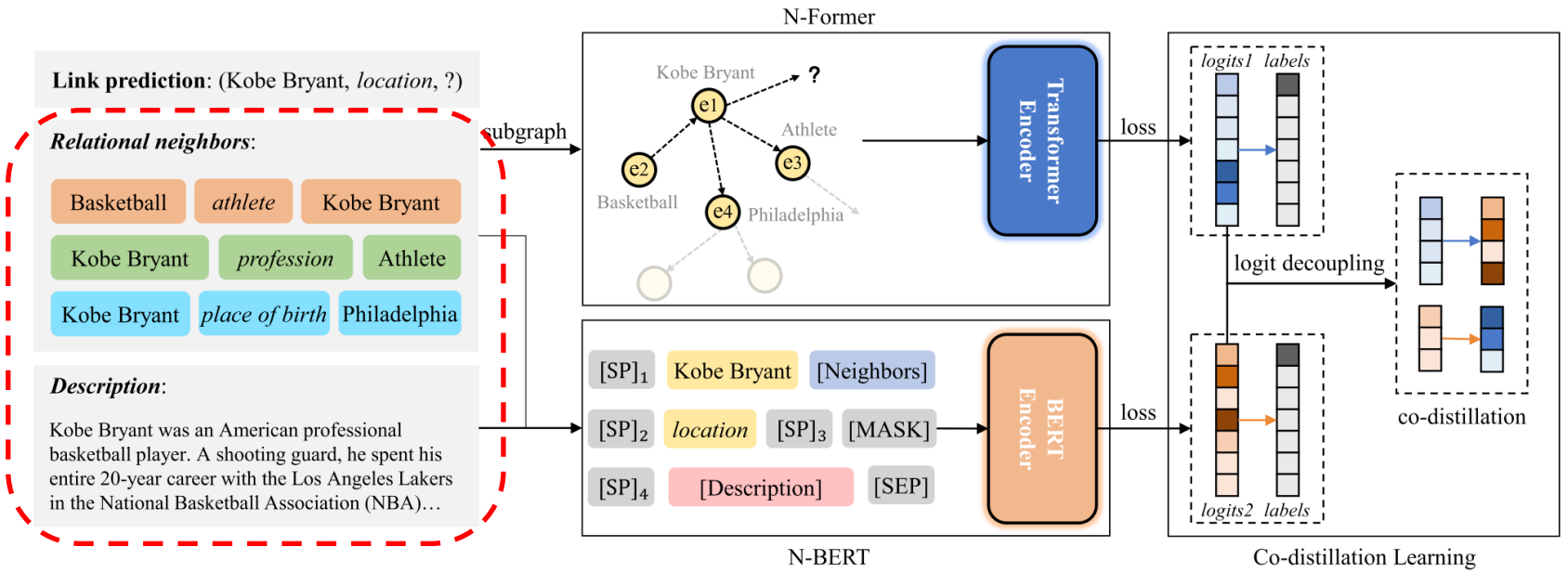
Method | framework

Constructing the input of models

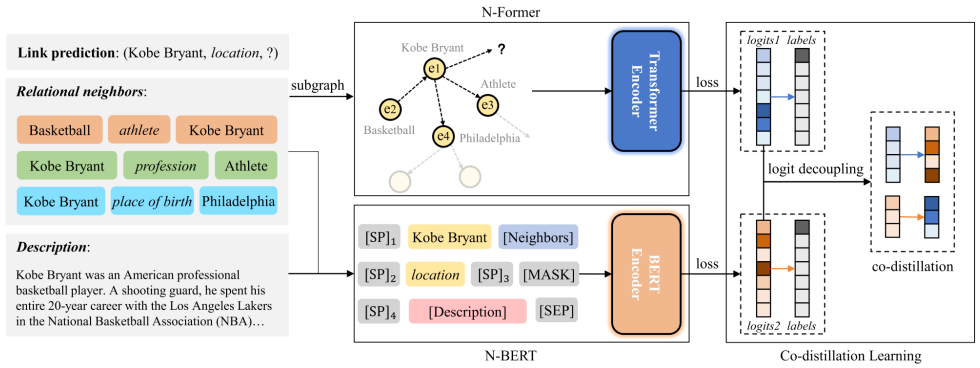
The available information to support this prediction includes

- relational neighbors
- textual descriptions

of Kobe Bryant



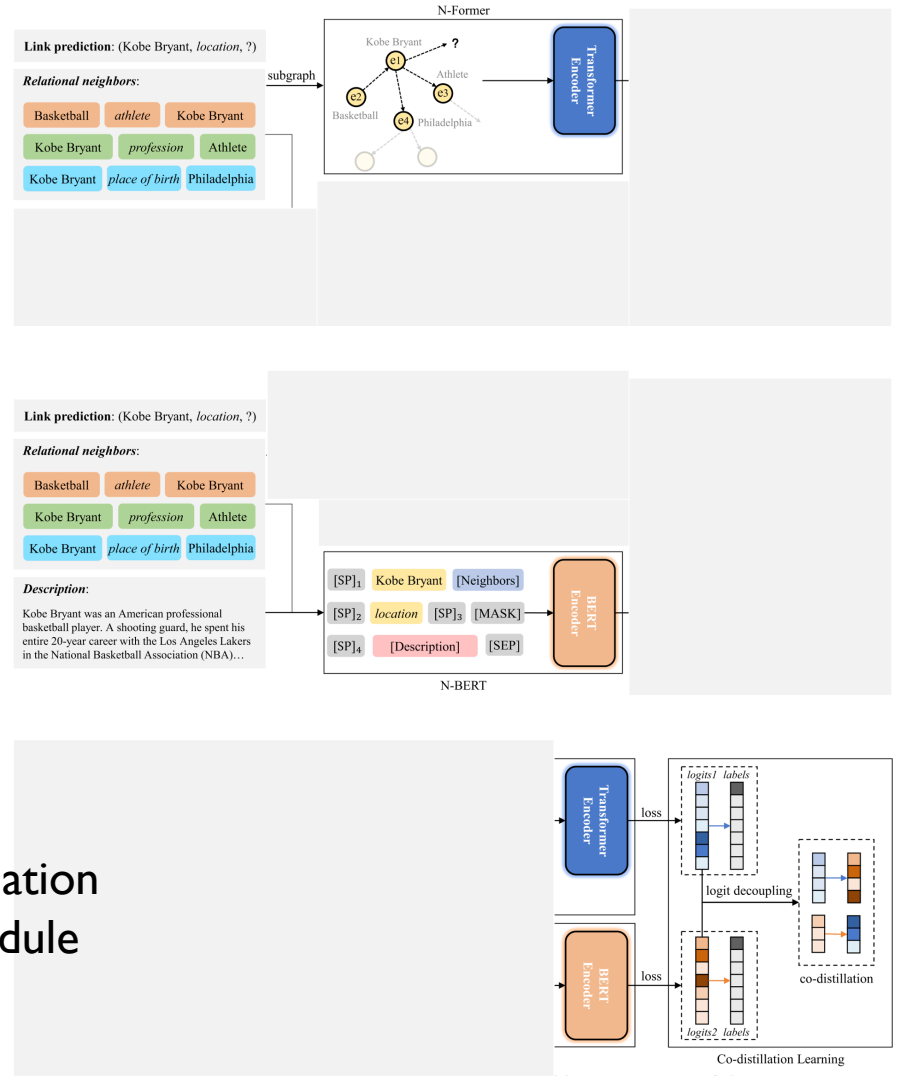
Method | framework



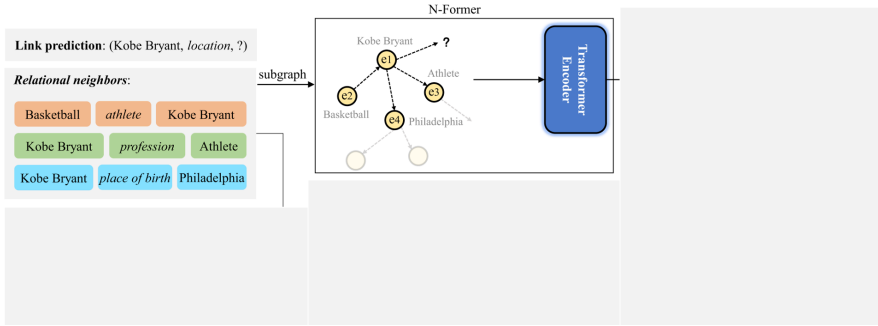
SLM

PLM

distillation module



Method | part1 N-Former



step 1. The N-Former first takes as input the subgraph to reconstruct a representation for Kobe Bryant.

$$\mathbf{E}^{S}_{Neighbor} = \sum_{(h',r')\in \mathtt{Neighbor}(h)} \mathtt{Transformer}(\mathbf{E}^{0}_{h'}, \mathbf{E}^{0}_{r'}, \mathbf{E}^{0}_{[\mathrm{MASK}]})$$

step 2. subgraph representation together with initial embedding and relation location are fed into N-Former

- to reconstruct a representation for the missing entity denoted as a special placeholder "[MASK]".

$$\mathbf{E}^{S}_{[\mathrm{MASK}]} = \mathtt{Transformer}(\mathtt{mean}(\mathbf{E}^{0}_{h}, \mathbf{E}^{S}_{Neighbor}), \mathbf{E}^{0}_{r}, \mathbf{E}^{0}_{[\mathrm{MASK}]})$$

step 3. The output "[MASK]" representation is used to compute the prediction logits.

$$\mathbf{P}_t = \mathtt{softmax}\left(\mathbf{E}_{\mathrm{ENT}} \cdot \mathtt{MLP}(\mathbf{E}_t)\right)$$

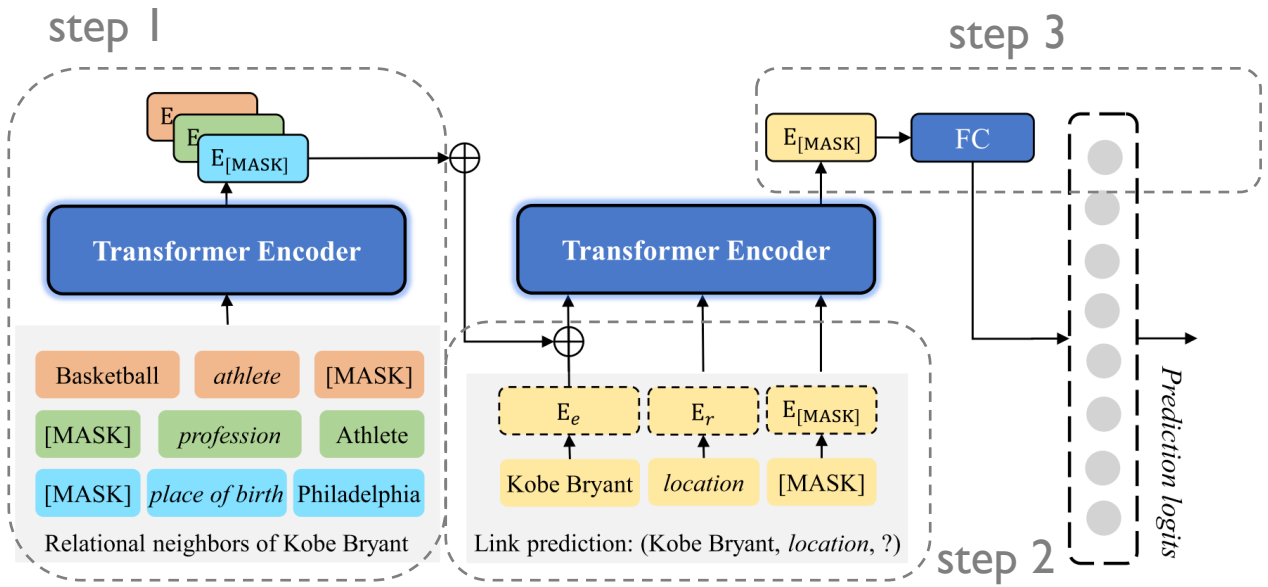
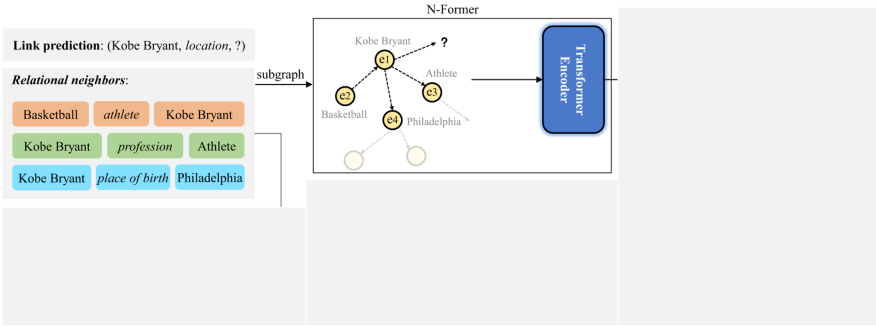


Figure 4: Architecture of the proposed N-Former.

Method | part1 N-Former



Optimization objectives

$$\mathbf{E}^{S}_{Neighbor} = \sum_{(h',r')\in \mathtt{Neighbor}(h)} \mathtt{Transformer}(\mathbf{E}^{0}_{h'}, \mathbf{E}^{0}_{r'}, \mathbf{E}^{0}_{[\mathrm{MASK}]})$$

$$\mathbf{E}^{S}_{[\mathrm{MASK}]} = \mathtt{Transformer}(\mathtt{mean}(\mathbf{E}^{0}_{h}, \mathbf{E}^{S}_{Neighbor}), \mathbf{E}^{0}_{r}, \mathbf{E}^{0}_{[\mathrm{MASK}]})$$

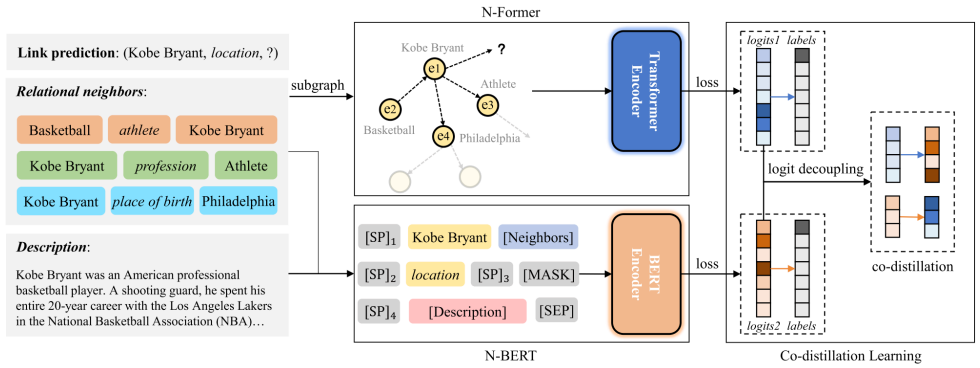
$$\mathcal{L}_{\mathrm{structure}} = \sum_{(h,r,t)\in\mathcal{T}} \left(\mathcal{L}_{\mathrm{triplet}}(t) + \mathtt{CrossEntropy}(\mathbf{P}^{S}_{t}, \mathbf{L}_{t})\right)$$

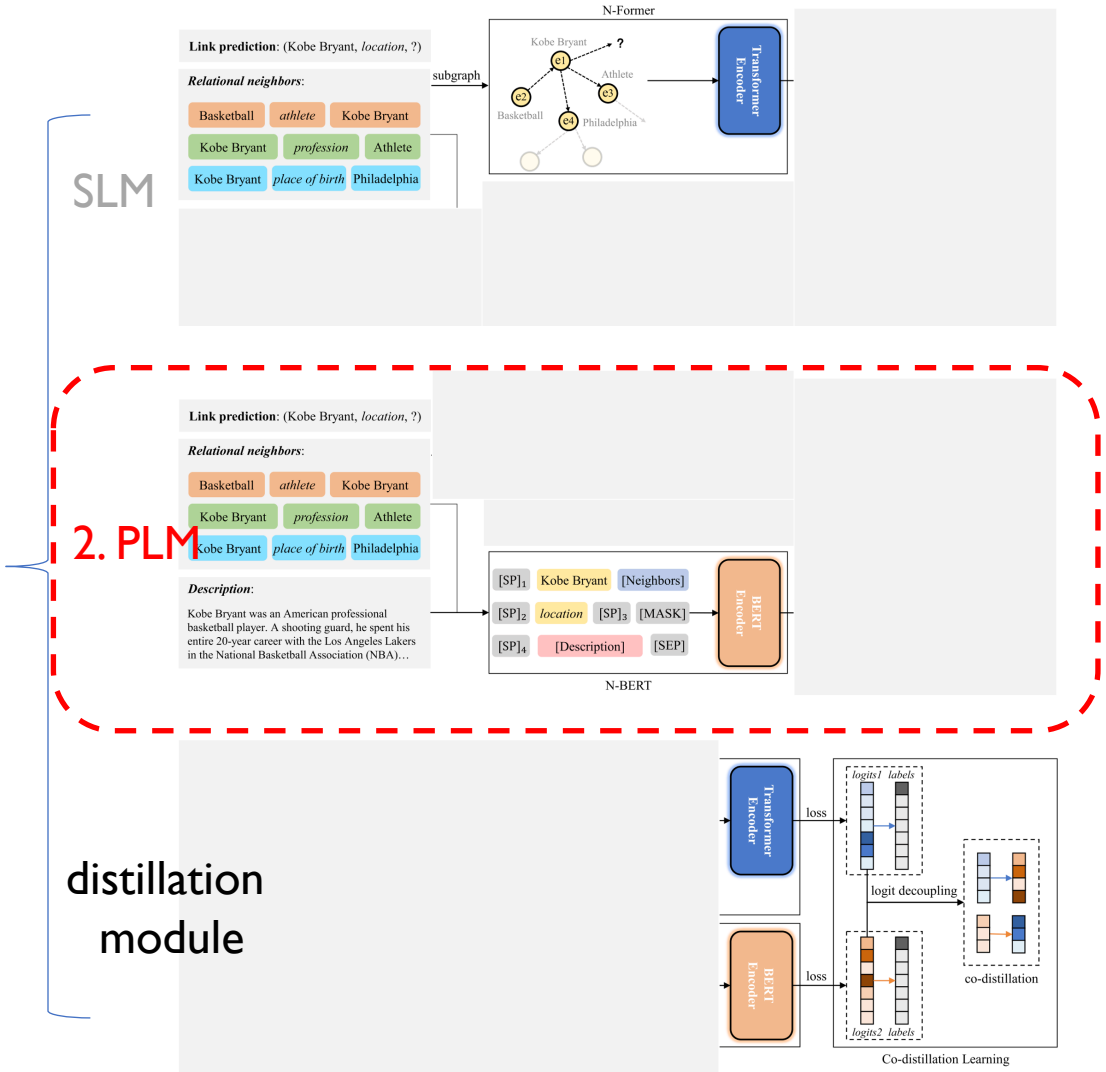
$$\mathbf{E}^{n}_{[\mathrm{MASK}]} = \mathtt{Transformer}(\mathbf{E}^{0}_{h}, \mathbf{E}^{0}_{r}, \mathbf{E}^{0}_{[\mathrm{MASK}]})$$

$$\mathbf{P}_{t} = \mathtt{softmax}\left(\mathbf{E}_{\mathrm{ENT}} \cdot \mathtt{MLP}(\mathbf{E}_{t})\right)$$

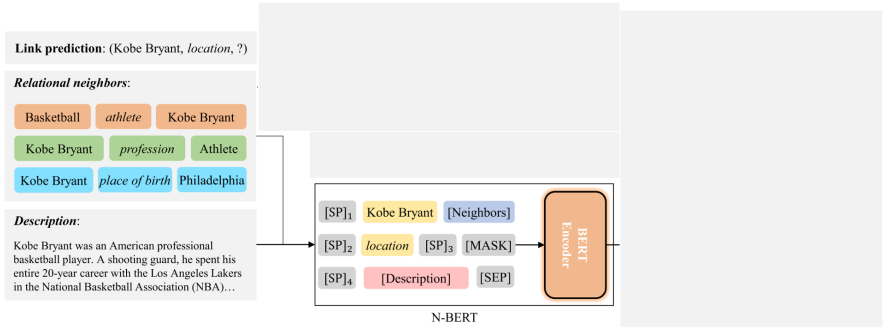
$$\mathcal{L}_{\mathrm{triplet}}(t) = \mathtt{CrossEntropy}(\mathbf{P}_{t}, \mathbf{L}_{t})$$

Method | framework





Method | part2 N-BERT



The key challenge for PLM-based link prediction

- is how to retrieve the relevant knowledge from PLMs

Given an incomplete triplet (h, r, [MASK]),
a typical solution is to introduce the textual description
of the entity for constructing a sentence-like prompt

For an incomplete triplet (h, r, [MASK]),
its prompt sentence is denoted by prompt(h, r)

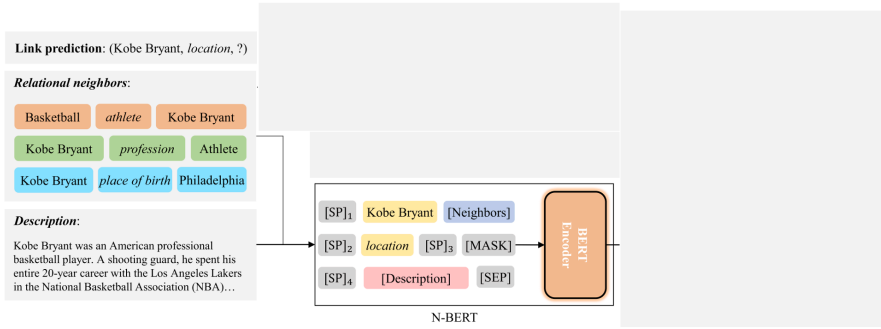
$$\mathbf{E}_{[\mathrm{MASK}]} = \mathrm{BERT}(\mathtt{prompt}_T(h, r)),$$

Table 1: Prompt templates used in N-BERT.

Triplet prompt	$\mathtt{prompt}_T(h,r) = [\mathrm{CLS}]\ N_\mathrm{h}\ [\mathrm{SEP}]\ N_\mathrm{r}\ [\mathrm{SEP}]\ [\mathrm{MASK}]\ [\mathrm{SEP}]\ D_\mathrm{h}\ [\mathrm{SEP}]$ $\mathtt{prompt}_T(r,t) = [\mathrm{CLS}]\ [\mathrm{MASK}]\ [\mathrm{SEP}]\ N_\mathrm{r}\ [\mathrm{SEP}]\ N_\mathrm{t}\ [\mathrm{SEP}]\ D_\mathrm{t}\ [\mathrm{SEP}]$
Relational prompt	$\mathtt{prompt}_R(h,r) = [\mathrm{CLS}]\ [\mathrm{SP}]_1^r\ N_\mathrm{h}\ [\mathrm{SP}]_2^r\ N_\mathrm{r}\ [\mathrm{SP}]_3^r\ [\mathrm{MASK}]\ [\mathrm{SP}]_4^r\ D_\mathrm{h}\ [\mathrm{SEP}]$ $\mathtt{prompt}_R(r,t) = [\mathrm{CLS}]\ [\mathrm{SP}]_1^r\ [\mathrm{MASK}]\ [\mathrm{SP}]_2^r\ N_\mathrm{r}\ [\mathrm{SP}]_3^r\ N_\mathrm{t}\ [\mathrm{SP}]_4^r\ D_\mathrm{t}\ [\mathrm{SEP}]$
Neighbor prompt	$\mathtt{prompt}_N(h,r) = [\mathrm{CLS}]\ [\mathrm{SP}]_1^r\ N_\mathrm{h}\ [\mathrm{Neighbors}]\ [\mathrm{SP}]_2^r\ N_\mathrm{r}\ [\mathrm{SP}]_3^r\ [\mathrm{MASK}]\ [\mathrm{SP}]_4^r\ D_\mathrm{h}\ [\mathrm{SEP}]$ $\mathtt{prompt}_N(r,t) = [\mathrm{CLS}]\ [\mathrm{SP}]_1^r\ [\mathrm{MASK}]\ [\mathrm{SP}]_2^r\ N_\mathrm{r}\ [\mathrm{SP}]_3^r\ N_\mathrm{t}\ [\mathrm{Neighbors}]\ [\mathrm{SP}]_4^r\ D_\mathrm{t}\ [\mathrm{SEP}]$

- [CLS]: classification
- [SEP]: separation
- [SP]: soft prompt

Method | part2 N-BERT



1. The PLM-based model N-BERT constructs a prompt from the descriptions, neighbors and names of entities as the input.

$$(\mathrm{h}, \mathrm{r}) \rightarrow \mathrm{prompt}(\mathrm{h}, \mathrm{r})$$

2. The prompt sentences would be fed into a BERT to get the representation of [MASK] w.r.t. the missing entity.

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

3. The output representation of BERT for "[MASK]" is used to predict the missing entity.

$$f\left(\mathbf{E}_{[\mathrm{MASK}]}\right) \rightarrow logits$$

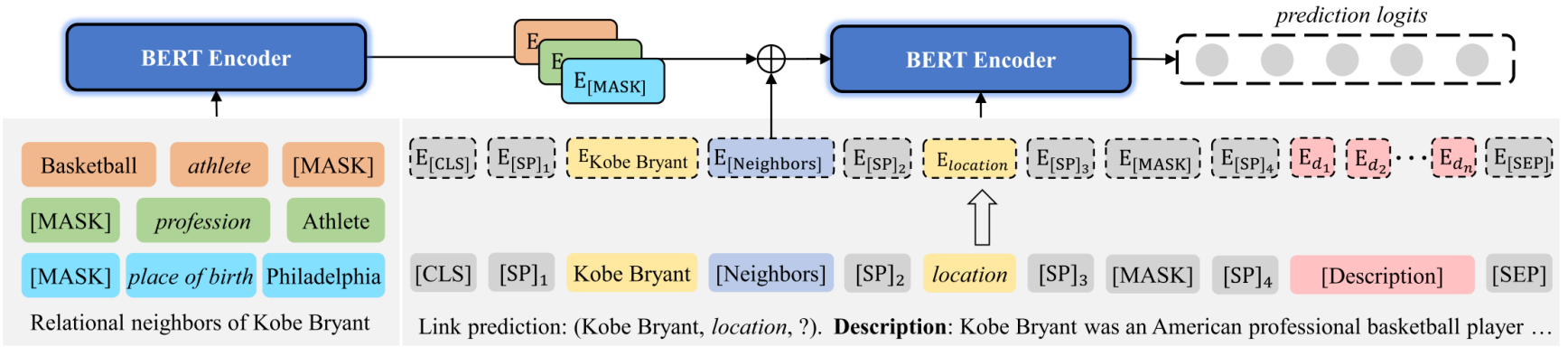
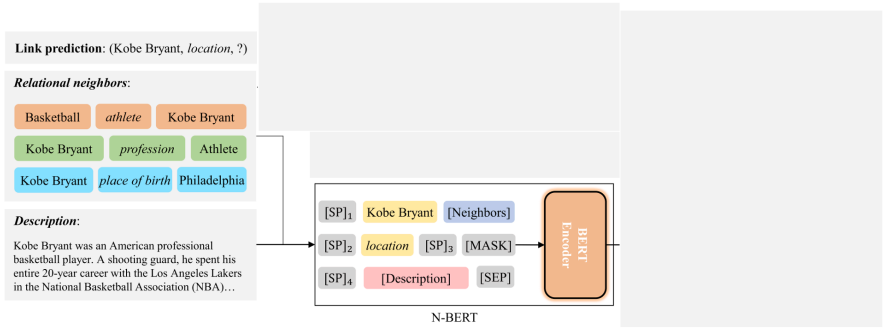


Figure 5: Architecture of the proposed N-BERT.

Method | part2 N-BERT



Some further details of N-BERT

- To make prompts more expressive and take full advantage of the PLMs

Table 1: Prompt templates used in N-BERT.

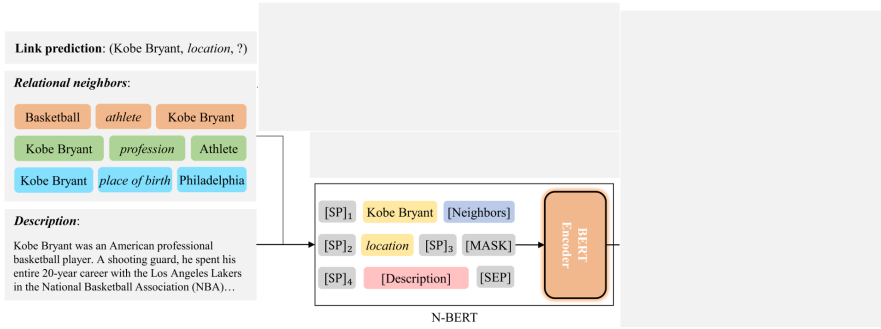
Triplet prompt	$\texttt{prompt}_T(h,r) = \text{[CLS]}\ N_\text{h}\ \text{[SEP]}\ N_\text{r}\ \text{[SEP]}\ \text{[MASK]}\ \text{[SEP]}\ D_\text{h}\ \text{[SEP]}$
	$\texttt{prompt}_T(r,t) = \text{[CLS]}\ \text{[MASK]}\ \text{[SEP]}\ N_\text{r}\ \text{[SEP]}\ N_\text{t}\ \text{[SEP]}\ D_\text{t}\ \text{[SEP]}$
Relational prompt	$\texttt{prompt}_R(h,r) = \text{[CLS]}\ [\text{SP}]_1^r\ N_\text{h}\ [\text{SP}]_2^r\ N_\text{r}\ [\text{SP}]_3^r\ \text{[MASK]}\ [\text{SP}]_4^r\ D_\text{h}\ \text{[SEP]}$
	$\texttt{prompt}_R(r,t) = \text{[CLS]}\ [\text{SP}]_1^r\ \text{[MASK]}\ [\text{SP}]_2^r\ N_\text{r}\ [\text{SP}]_3^r\ N_\text{t}\ [\text{SP}]_4^r\ D_\text{t}\ \text{[SEP]}$
Neighbor prompt	$\texttt{prompt}_N(h,r) = \text{[CLS]}\ [\text{SP}]_1^r\ N_\text{h}\ \text{[Neighbors]}\ [\text{SP}]_2^r\ N_\text{r}\ [\text{SP}]_3^r\ \text{[MASK]}\ [\text{SP}]_4^r\ D_\text{h}\ \text{[SEP]}$
	$\texttt{prompt}_N(r,t) = \text{[CLS]}\ [\text{SP}]_1^r\ \text{[MASK]}\ [\text{SP}]_2^r\ N_\text{r}\ [\text{SP}]_3^r\ N_\text{t}\ \text{[Neighbors]}\ [\text{SP}]_4^r\ D_\text{t}\ \text{[SEP]}$

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

$$\mathbf{E}^T_{Neighbor} = \sum_{(h',r') \in \texttt{InNeighbor}(h)} \texttt{BERT}(\texttt{prompt}_R(h',r')) + \sum_{(r',t') \in \texttt{OutNeighbor}(h)} \texttt{BERT}(\texttt{prompt}_R(r',t'))$$

$$\mathbf{E}^T_t = \texttt{BERT}(\texttt{prompt}_N(h,r))$$

Method | part2 N-BERT



Optimization objectives

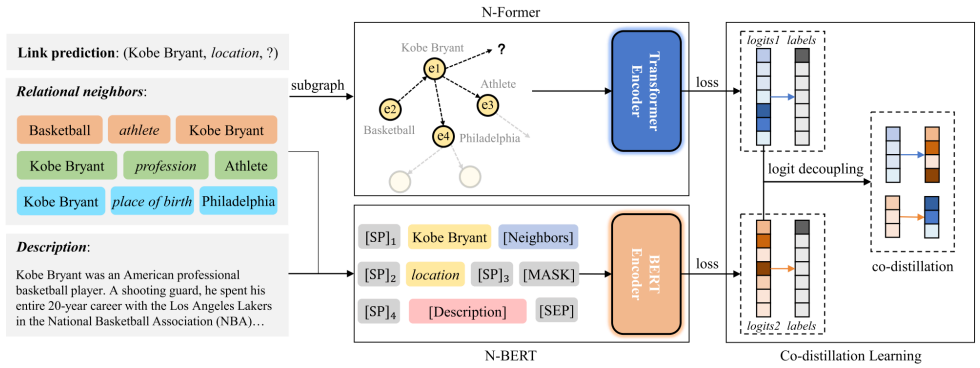
For a training triple (h, r, t)

Head prediction (?, r, t) $\mathbf{E}_h^T = \mathrm{BERT}(\mathrm{prompt}_N(r,t))$

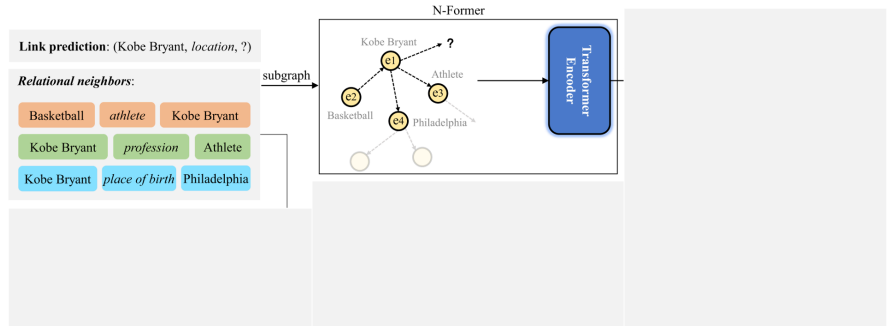
Tail prediction (h, r, ?) $\mathbf{E}_t^T = \mathrm{BERT}(\mathrm{prompt}_N(h,r))$

$$\Rightarrow \quad \mathcal{L}_{\text{text}} = \sum_{(h,r,t)\in\mathcal{T}} \left(\mathrm{CrossEntropy}(\mathbf{P}_h^T, \mathbf{L}_h) + \mathrm{CrossEntropy}(\mathbf{P}_t^T, \mathbf{L}_t)\right)$$

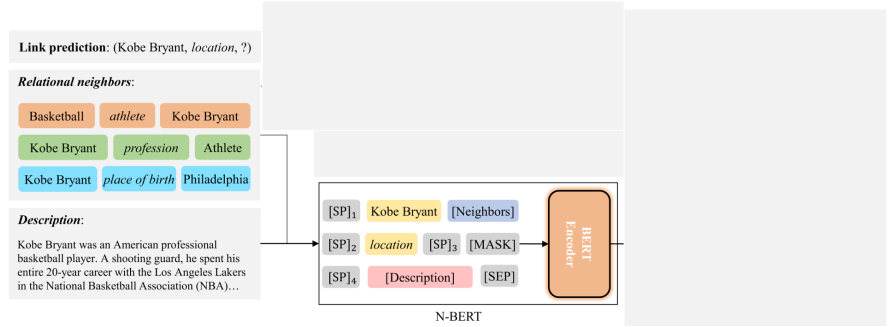
Method | framework



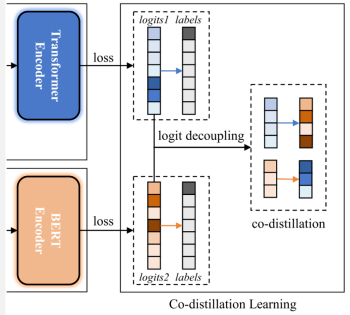
SLM



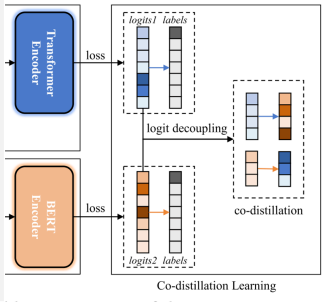
PLM



3. distillation module



Method | part3 distillation



A knowledge co-distillation method

- assume two models are complementary, they should teach each other what they are good at

Co-distillation:
a model can be the teacher
and the student at the same time

(picked from the paper):

For our N-Former and N-BERT, we empirically find that the scores for target entities both become higher in the training process.

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.

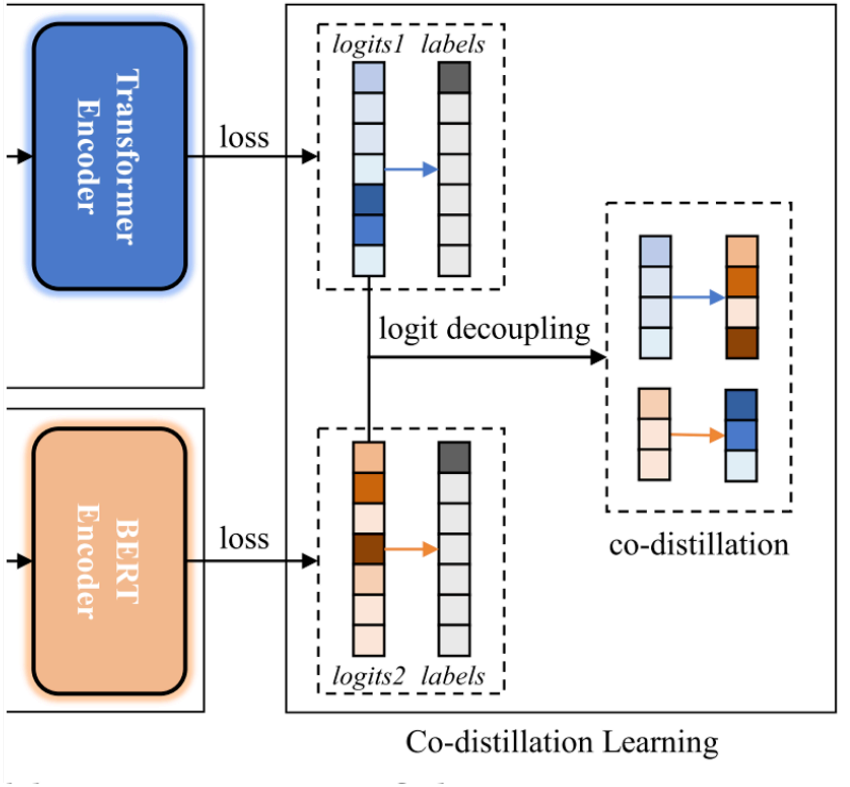
Method | part3 distillation

Co-distillation

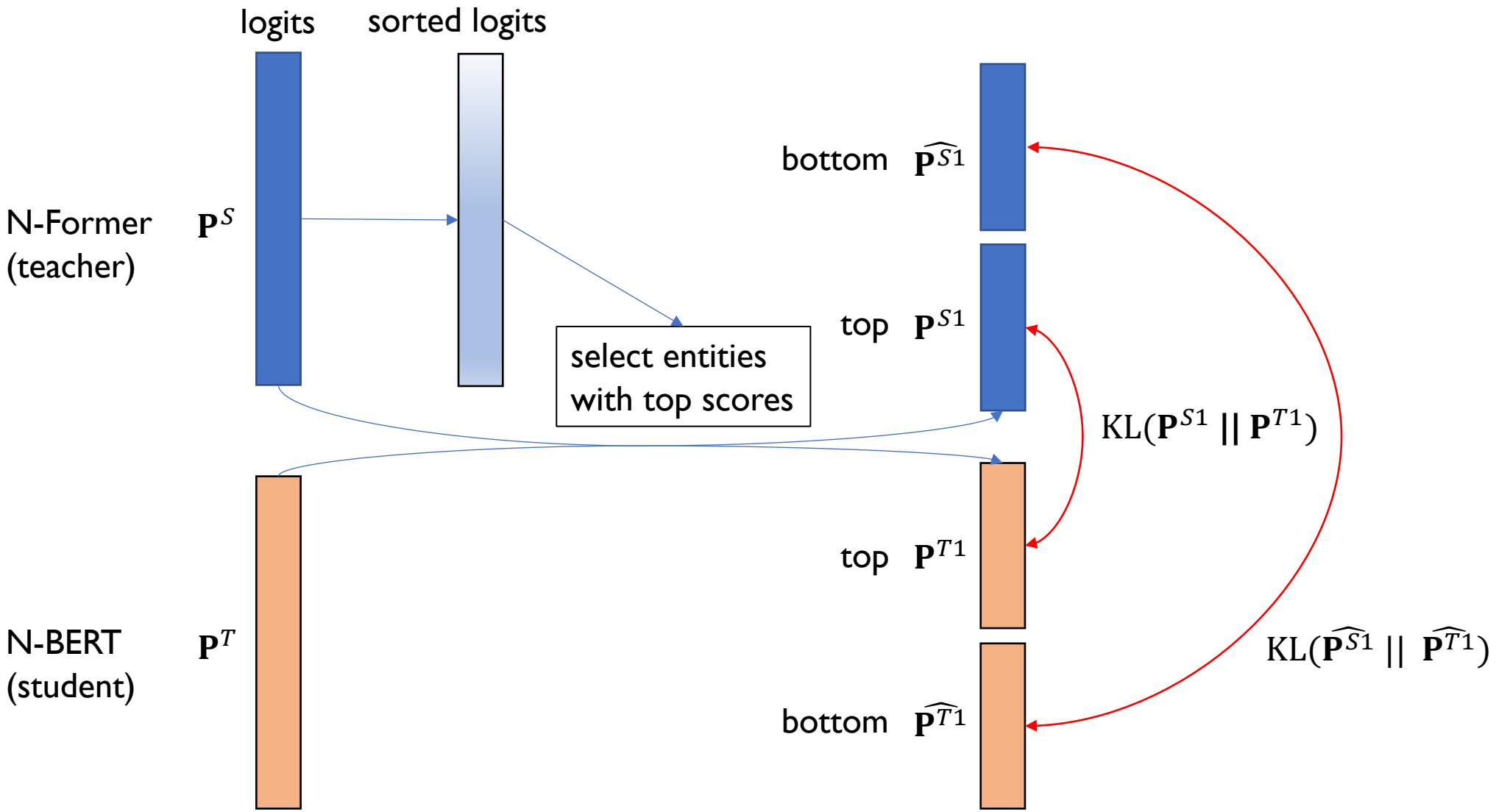
1. divide the logits into two parts,
one concerning the high-confidence predictions of N-Former
and the other for N-BERT.

2. enable bidirectional knowledge transfer
by minimizing the KL divergence
of the partial logits from the two models.

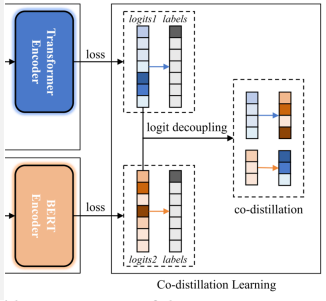
$$\mathcal{L}_{KD}(\mathbf{P}_t^{S_1}, \mathbf{P}_t^{T_1}) = \mathrm{KL}(\mathbf{b}_t^{S_1} \,||\, \mathbf{b}_t^{T_1}) + \mathrm{KL}(\hat{\mathbf{P}}_t^{S_1} \,||\, \hat{\mathbf{P}}_t^{T_1})$$



Method | part3 distillation



Method | part3 distillation



Co-distillation

select half entities with higher scores and then compare via KL:

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

objectives for N-BERT and N-Former:

$$\mathcal{L}_{\text{N-BERT}} = \alpha\, \mathcal{L}_{KD}(\mathbf{P}_t^{S_1}, \mathbf{P}_t^{T_1}) + (1-\alpha)\, \texttt{CrossEntropy}(\mathbf{P}_t^{T}, \mathbf{L}_t)$$

$$\mathcal{L}_{\text{N-Former}} = \beta\, \mathcal{L}_{KD}(\mathbf{P}_t^{T_2}, \mathbf{P}_t^{S_2}) + (1-\beta)\, \texttt{CrossEntropy}(\mathbf{P}_t^{S}, \mathbf{L}_t).$$

Method | full algorithm

Eq. (6):

$$\mathcal{L}_{\text{structure}} = \sum_{(h,r,t)\in\mathcal{T}} \left(\mathcal{L}_{\text{triplet}}(t) + \texttt{CrossEntropy}(\mathbf{P}_t^S, \mathbf{L}_t)\right)$$

Eq. (11):

$$\mathcal{L}_{\text{text}} = \sum_{(h,r,t)\in\mathcal{T}} \left(\texttt{CrossEntropy}(\mathbf{P}_h^T, \mathbf{L}_h) + \texttt{CrossEntropy}(\mathbf{P}_t^T, \mathbf{L}_t)\right)$$

Eq. (14):

$$\mathcal{L}_{\text{N-BERT}} = \alpha\, \mathcal{L}_{KD}(\mathbf{P}_t^{S_1}, \mathbf{P}_t^{T_1}) + (1-\alpha)\, \texttt{CrossEntropy}(\mathbf{P}_t^T, \mathbf{L}_t)$$

$$\mathcal{L}_{\text{N-Former}} = \beta\, \mathcal{L}_{KD}(\mathbf{P}_t^{T_2}, \mathbf{P}_t^{S_2}) + (1-\beta)\, \texttt{CrossEntropy}(\mathbf{P}_t^S, \mathbf{L}_t).$$

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 T_train;
for epoch ← 1 to max_epoch_num do
    for step ← 1 to max_step_num do
        b ← 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 T_valid;
    if early stop then break;
Test CoLE using T_test;
```

Outline

- Background
- The proposed method
- Experiment
- Summary and discussion

Experiment

Co-distillation can boost
both N-Former and N-BERT,
result in higher individual performances

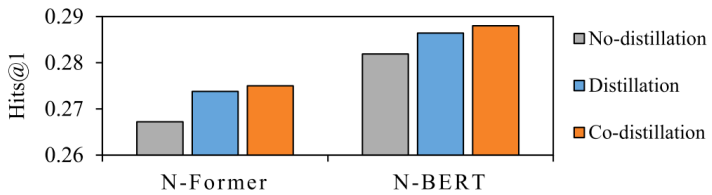


Figure 8: Hits@1 results of our models when using no-distillation, the conventional (unidirectional) distillation and our co-distillation on FB15K-237.

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^2GCN	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**

Experiment

Ensemble N-Former and N-BERT can be better

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**	0.530	**0.609**	0.692	0.585
$\text{N-Former}_{\text{co-distilled}}$	0.279	0.412	0.556	0.373	0.446	0.504	0.581	0.489
$\text{N-BERT}_{\text{co-distilled}}$	0.293	0.426	0.570	0.387	**0.532**	0.607	0.689	0.585
$\text{ProbsMax}_{\text{co-distilled}}$	0.292	0.428	0.571	0.387	0.518	0.595	0.684	0.574
$\text{ProbsAvg}_{\text{co-distilled}}$	**0.294**	**0.430**	**0.574**	**0.389**	**0.532**	0.608	**0.694**	**0.587**

The proposed designs for N-Former / N-BERT are effective

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
$\text{N-Former}_{\text{w/o neighbor}}$	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
$\text{N-BERT}_{\text{w/o soft prompt}}$	0.264	0.389	0.527	0.354	0.510	0.579	0.672	0.563
$\text{N-BERT}_{\text{w/o description}}$	0.271	0.408	0.556	0.368	0.414	0.488	0.571	0.467
$\text{N-BERT}_{\text{w/o neighbor}}$	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

Experiment

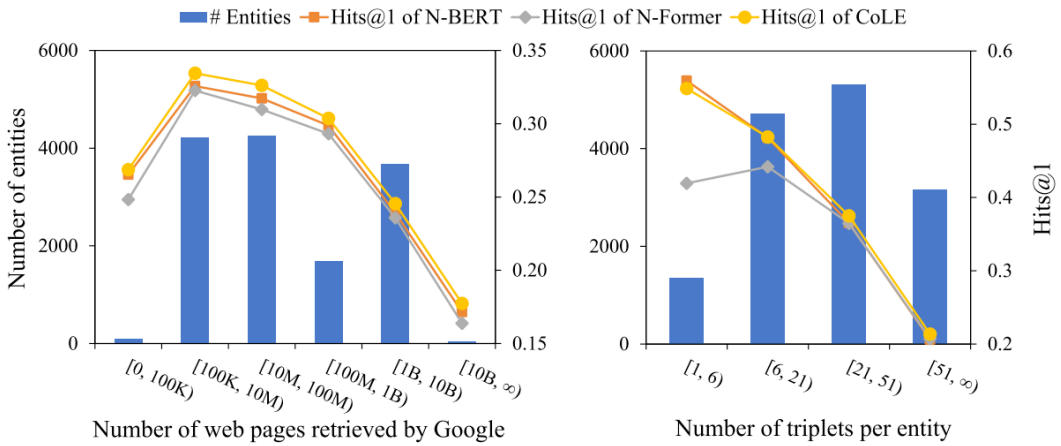
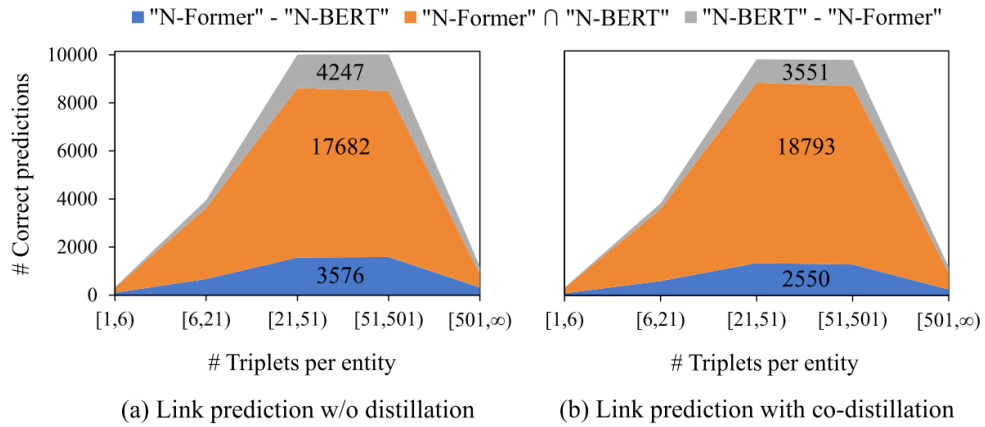


Figure 7: Hits@1 results on FB15K-237 along with the numbers of retrieved web pages and edges per entity, resp.



(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.

Co-distillation leverages the complementarity and transfers the unique knowledge of each model mutually, thus benefiting them.

Outline

- Background
- The proposed method
- Experiment
- Summary and discussion

Summary

1. (methodology) reveal that PLMs and SLMs can be complementary.
2. (model) adopt PLMs and SLMs for KG reasoning with adaptive designs,
   especially the construction of model input that is dependent on the specific query.
3. (optimization) a co-distillation method for transferring knowledge among PLMs and SLMs.
4. (experiment) good performances on WN18RR/FB15k-237.

Potential directions

(model design): information injecting and utilizing

1. The idea and adaptive designs of "query-dependent"

(learning paradigm): knowledge transferring

2. Knowledge distillation of PLMs and SLMs

(interpretability): knowledge representing

3. Interpreting SLMs via PLMs

Potential directions | "query-dependent"

The high-level idea of "query-dependent" and the corresponding adaptive designs

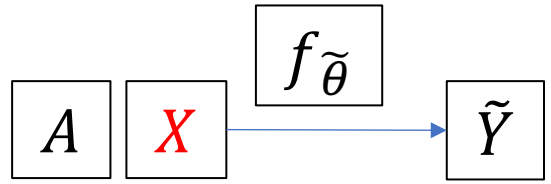
- current query-dependent SLMs/PLMs may not fully utilize the graph structure

in SLMs like RED-GNN or NBFNet,

- only the head entity is assigned effective embedding before propagation

However,

- its propagation is strongly-correlated with the location of head entity

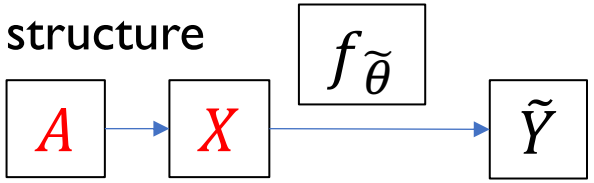


in PLMs like N-BERT or KG-BERT,

- the prompt templates are specially designed w.r.t. queries and the graph structure

However,

- the input templates are manually designed and fixed for each query
- the neighboring information is compressed to a fixed-size embedding of token



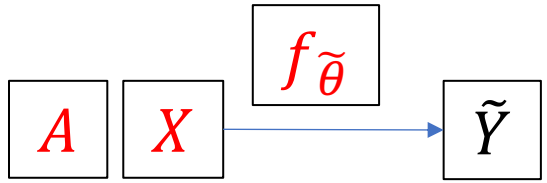
Potential directions | "query-dependent"

So, is there a better way to a KG reasoning model? 🤔

- in addition to utilizing the local information in KG
- SLMs → long-range information (internal of KG)
- PLMs → context information (external of KG)

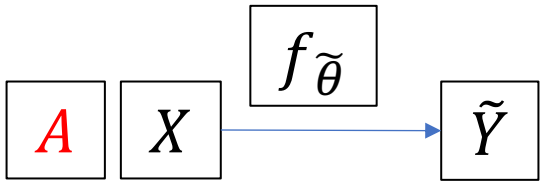
Maybe, we can consider a query-dependent transformer?

- jump out of the local area of head entity
  - can potentially explore the full KG
- injecting the token embeddings of PLMs as inputs
  - as initial entity embeddings that are also inductive



Potential directions | "query-dependent"

Refining the observed graph



dataset	# train triples	# validate triples	# test triples
WN18RR	17368	5562	5716
FB15k237	54424	20112	22850
NELL995	29936	802	3639

dataset	setting	Test MRR	Test H@1	Test H@10
WN18RR	w/o validate triples	0.5649	0.4992	0.6793
	add validate triples	0.5856	0.5230	0.6961
FB15k237	w/o validate triples	0.4177	0.3315	0.5852
	add validate triples	0.4044	0.3186	0.5685
NELL995	w/o validate triples	0.5651	0.5012	0.6686
	add validate triples	0.5659	0.5023	0.6702

Potential directions | "knowledge"

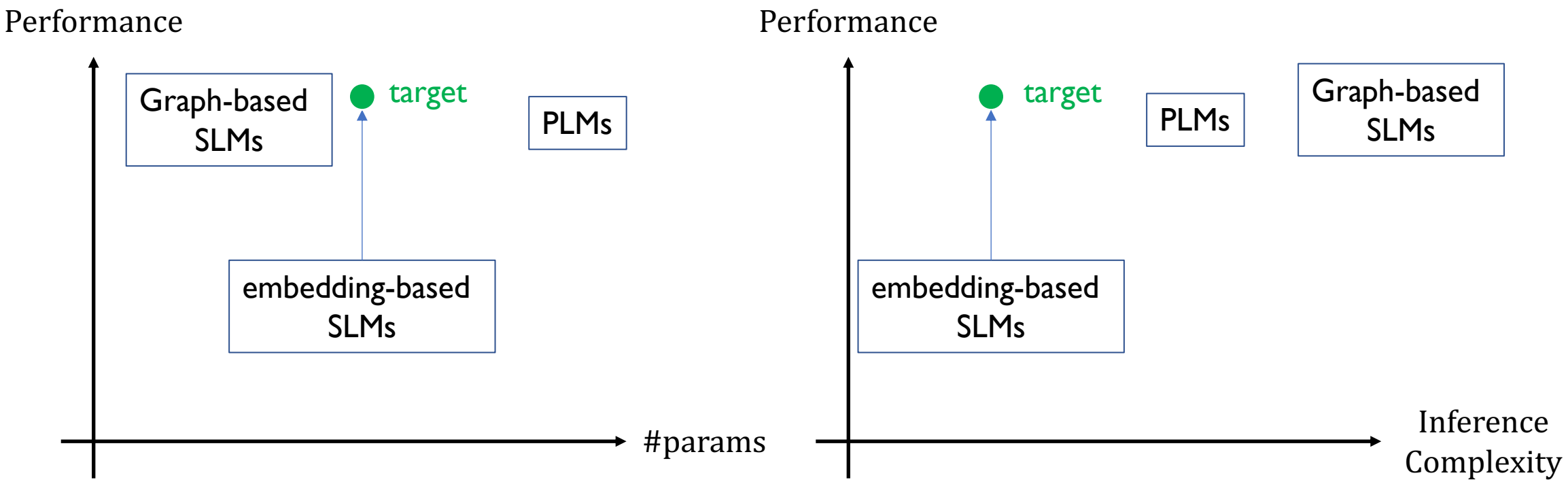
Extract and transfer the knowledge among PLMs and SLMs

- the large-scale PLMs are informative and the GNN-based SLMs are powerful
  - however, they are complex and expensive whether in training or inference

- so, can their knowledge be extracted and stored in the more efficient model?
  - e.g., embedding models
  - however, embedding models can not explicitly utilize the KG structure

So, can we ensemble all their advantages in one model? 🤔
➔ The knowledge distillation technique is a natural solution here.

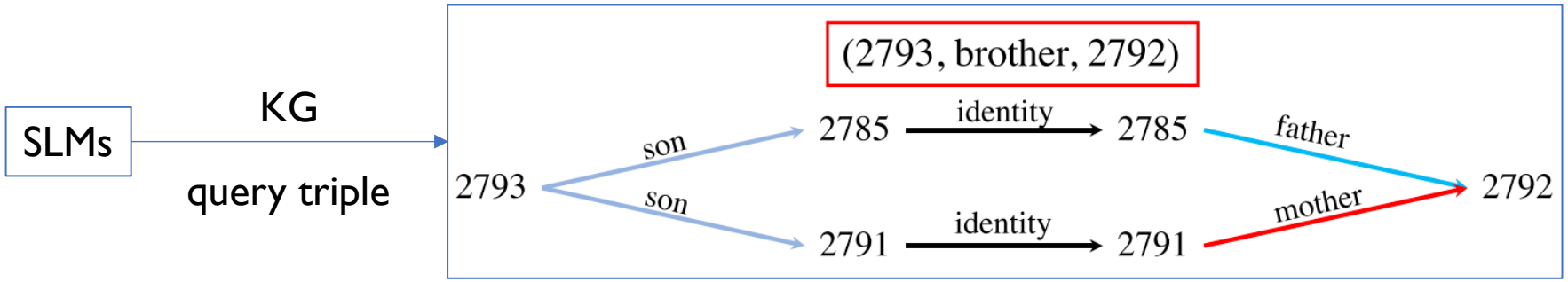
Potential directions | "knowledge"



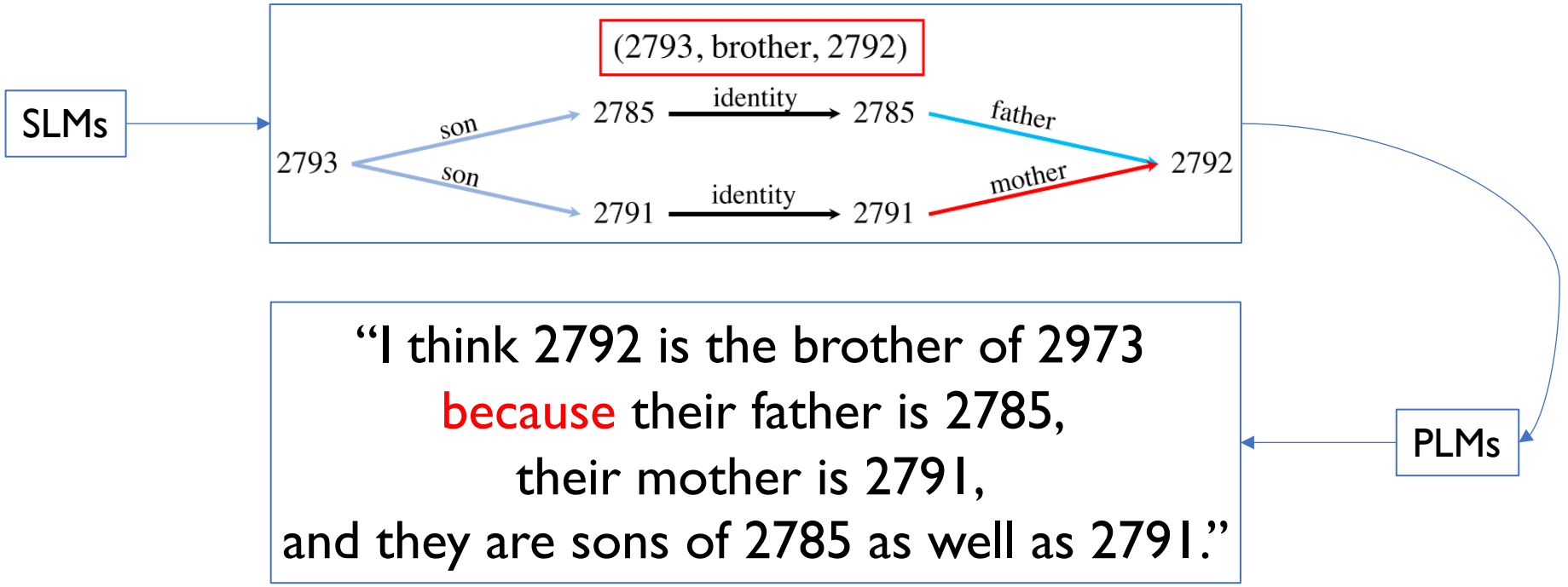
However, existing distillation works are mainly for models of similar architectures

- e.g., distillate a 18-layer resnet from a 101-layer resnet
- distillation across different models is still under-explored 🤔

Potential directions | "interpretable"



Potential directions | "interpretable"

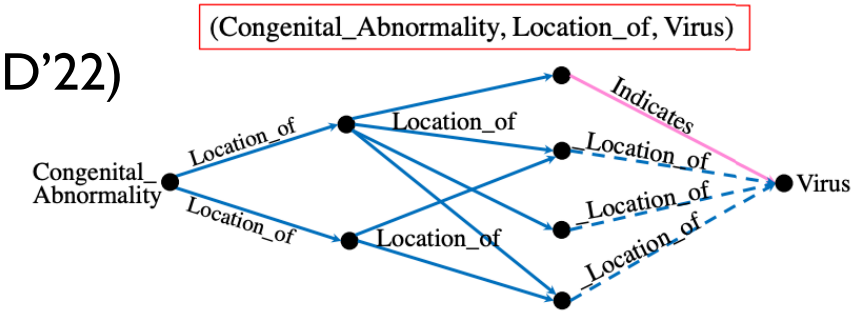


➔ Can we use PLMs to generate sentences for interpreting? 🤔

Potential directions | "interpretable"

The latest methods for interpreting the SLMs

- can tell which triples are most important for a query (SIGMOD'22)
  - in a post-hoc manner for interpreting embedding models
- or subgraphs for GNN-based model (WWW'21)
  - but the subgraph size is hard to control



But, the weighted triples/subgraphs might still not easy to be understood

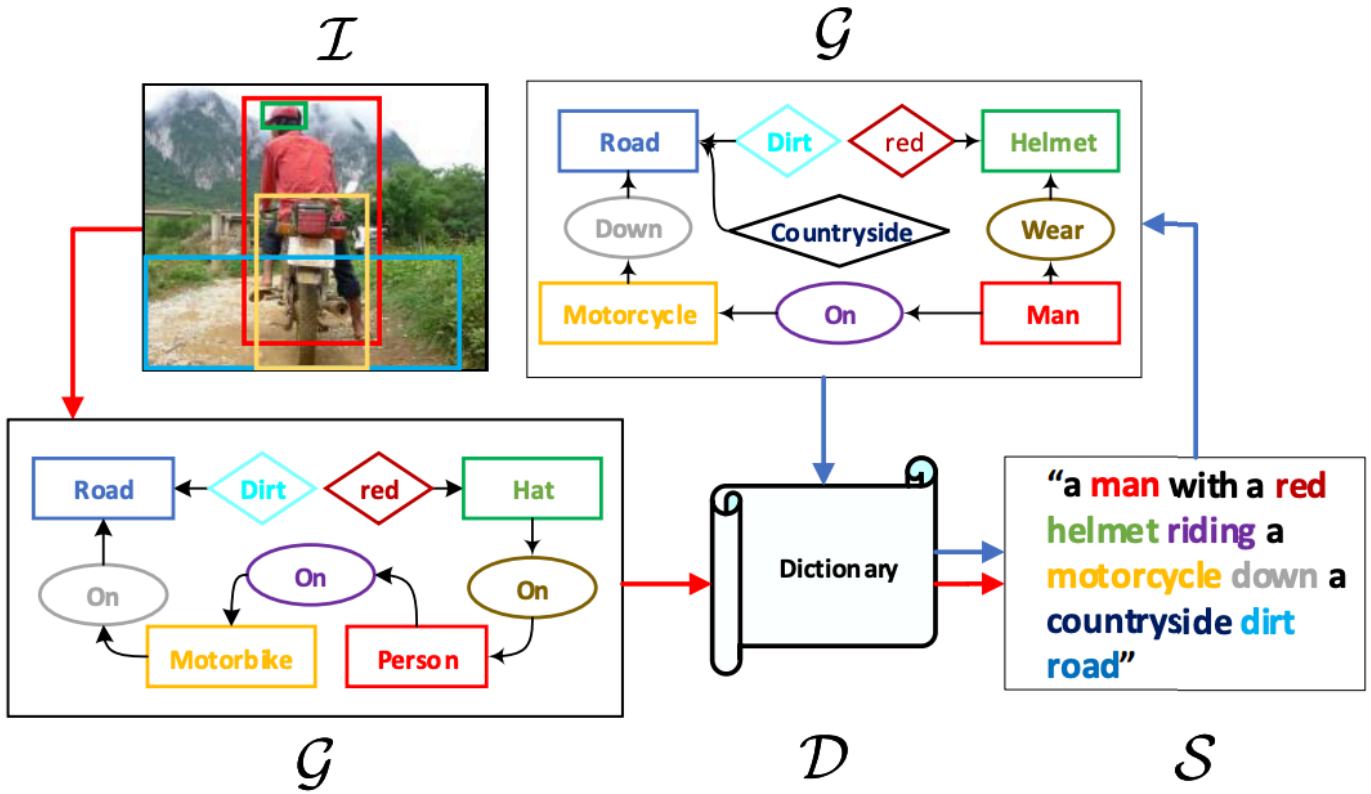
- understanding the underneath correlation among them is hard for human
- especially when important triples are quite a lot

So, can we utilize the PLMs for interpreting the SLMs with natural languages?

- i.e., watch the weighted triples (or subgraphs), and summarize them into sentences.
- i.e., 看"图"说"话"

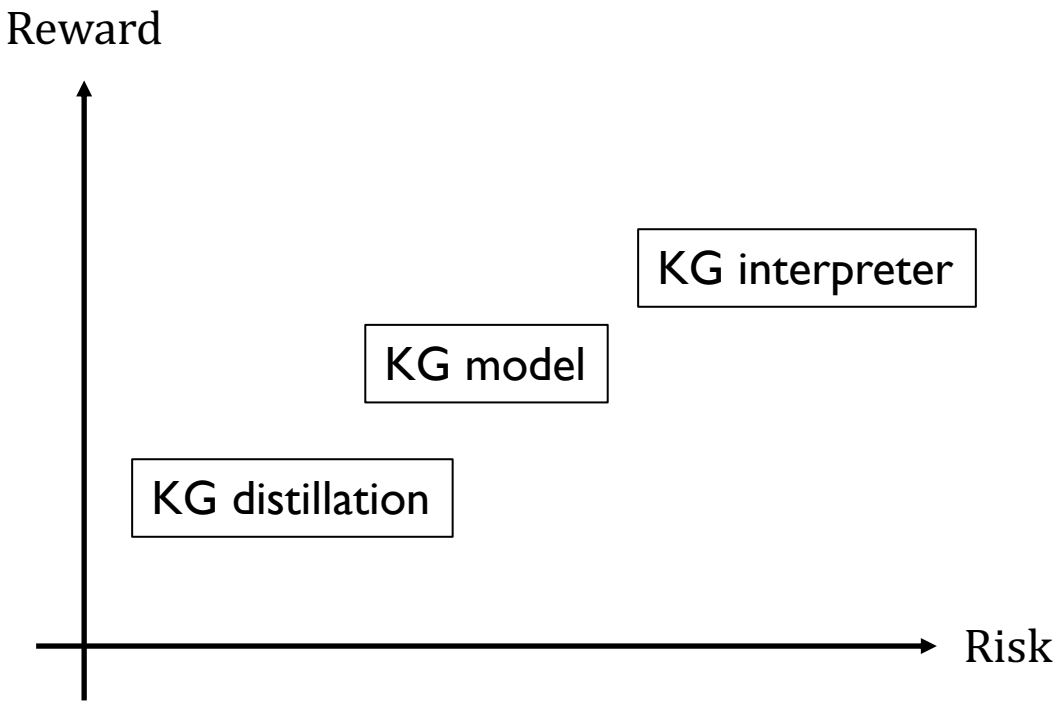
Potential directions | "interpretable"

Scene graph captioning



Auto-Encoding Scene Graphs for Image Captioning. 2018

Potential directions | summary



Related works | PLMs for KG learning

1. Language Models as Knowledge Bases? EMNLP 2019.
2. Pretrain-KGE: Learning Knowledge Representation from Pretrained Language Models. EMNLP 2020.
3. KG-BERT: BERT for Knowledge Graph Completion. AAAI 2020.
4. Scientific Language Models for Biomedical Knowledge Base Completion: An Empirical Study. AKBC 2020.
5. MLMLM: Link Prediction with Mean Likelihood Masked Language Model. ACL-IJCNLP 2021.
6. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. ACM Transactions on Computing for Healthcare 2021.
7. I Know What You Do Not Know: Knowledge Graph Embedding via Co-distillation Learning. CIKM 2022.
8. Structure-Augmented Text Representation Learning for Efficient Knowledge Graph Completion. WWW 2021.
9. Do Pre-trained Models Benefit Knowledge Graph Completion? A Reliable Evaluation and a Reasonable Approach. ACL 2022.
10. SimKGC: Simple Contrastive Knowledge Graph Completion with Pre-trained Language Models. ACL 2022.
11. Reasoning Through Memorization: Nearest Neighbor Knowledge Graph Embeddings. Arxiv 2022.

Q&A

Thanks for your attention!