Incorporating Explicit Knowledge in Pre-trained Language Models for Passage Re-ranking

Qian Dong[*]
dongqian19@mails.ucas.ac.cn
Institute of Software, Chinese Academy of Sciences & University of Chinese Academy of Sciences
Beijing, China

Yiding Liu
Suqi Cheng
liuyiding.tanh@gmail.com
chengsuqi@gmail.com
Baidu Inc.
Beijing, China

Shuaiqiang Wang
Zhicong Cheng
shqiang.wang@gmail.com
chengzhicong01@baidu.com
Baidu Inc.
Beijing, China

Shuzi Niu[†]
shuzi@iscas.ac.cn
Institute of Software, Chinese Academy of Sciences
Beijing, China

Dawei Yin[†]
yindawei@acm.org
Baidu Inc.
Beijing, China

ABSTRACT

Passage re-ranking is to obtain a permutation over the candidate passage set from retrieval stage. Re-rankers have been boomed by Pre-trained Language Models (PLMs) due to their overwhelming advantages in natural language understanding. However, existing PLM based re-rankers may easily suffer from vocabulary mismatch and lack of domain specific knowledge. To alleviate these problems, explicit knowledge contained in knowledge graph is carefully introduced in our work. Specifically, we employ the existing knowledge graph which is incomplete and noisy, and first apply it in passage re-ranking task. To leverage a reliable knowledge, we propose a novel knowledge graph distillation method and obtain a knowledge meta graph as the bridge between query and passage. To align both kinds of embedding in the latent space, we employ PLM as text encoder and graph neural network over knowledge meta graph as knowledge encoder. Besides, a novel knowledge injector is designed for the dynamic interaction between text and knowledge encoder. Experimental results demonstrate the effectiveness of our method especially in queries requiring in-depth domain knowledge.

CCS CONCEPTS

• **Information systems** → **Language models**; **Learning to rank**; **Similarity measures**; *Novelty in information retrieval.*

KEYWORDS

Learning to Rank; Language models; Semantic Matching

[*]This work wad done during Qian Dong's internship at Baidu.
[†]Co-corresponding authors.

ACM Reference Format:
Qian Dong, Yiding Liu, Suqi Cheng, Shuaiqiang Wang, Zhicong Cheng, Shuzi Niu, and Dawei Yin . 2022. Incorporating Explicit Knowledge in Pre-trained Language Models for Passage Re-ranking. In *Proceedings of the 45th Int'l ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3477495.3531997

1 INTRODUCTION

Passage Re-ranking is a crucial stage in modern information retrieval systems, which aims to reorder a small set of candidate passages to be presented to users. To put the most relevant passages on top of a ranking list, a re-ranker is usually designed with powerful capacity in modeling semantic relevance, which attracted a wealth of research studies in the past decade [12]. Recently, large-scale pre-trained language models (PLMs), e.g. BERT [4], ERNIE [42] and RoBERTa [26], have dominated many natural language processing tasks, and have also achieved remarkable success on passage re-ranking. For example, PLM based re-rankers [5, 6, 23, 28] have achieved state-of-the-art performance, which takes the concatenation of query-passage pair as input, and applies multi-layer full-attention to model their semantic relevance. Their superiority can be attributed to the expressive transformer structure and the pretrain-then-finetune paradigm, which allow the model to learn useful implicit knowledge (i.e., semantic relevance in the latent space) from massive textual corpus [8].

However, *implicit knowledge* still has some inherent weaknesses, which limits the applicability of PLMs based re-rankers. First, queries and passages are usually created by different persons and have different expression ways [33], such as word usage and language style. Worse still, the data distributions of search queries and web contents are highly heterogeneous [25], where various specialized domains (e.g., bio-medical) may only have few training examples in a general corpus. Domain-specific knowledge can hardly be revealed and captured by the model, and thus the processing of domain-specific queries is often inaccurate.

To overcome the limitations, it is essential to incorporate the knowledge graph as *explicit knowledge* to PLM based re-rankers.

Thus we propose **K**nowledge **E**nhanced **R**e-ranking **M**odel (**KERM**), which utilizes external knowledge to *explicitly* enhance the semantic matching process in PLM based re-rankers. Intuitively, the difference in expression ways can be mitigated by the triplet with "synonymy" as relation in knowledge graph, and all the triplets can enrich the domain knowledge. The overall workflow of KERM is depicted in Fig. 1. To the the best of our knowledge, this is the first attempt for knowledge enhanced PLMs for passage re-ranking.

Despite the knowledge graph is a desirable source of explicit knowledge, it is non-trivial to take advantage of explicit knowledge directly for passage re-ranking due to the following two challenges:

- **Challenge 1.** Existing knowledge graph are not constructed for re-ranking task. They usually contain trivial factual triples, which can hardly bring information gain. The inappropriate selection of external knowledge could even jeopardize the re-ranker performance. How to utilize existing knowledge graph to re-ranking task is remain a challenge.
- **Challenge 2.** The explicit knowledge and implicit knowledge are highly heterogeneous due to the different sources, which makes the aggregation of the two difficult. How to mutually refine each other and effectively aggregate explicit knowledge into implicit knowledge to alleviate the semantic gap between query and passage is still a challenge.

In general, the workflow of KERM can be divided into knowledge graph distillation and knowledge aggregation to tackle the above challenges.

For *knowledge graph distillation*, we propose a novel pipeline to establish knowledge meta graphs, which only retain informative knowledge for passage re-ranking. Specifically, we first distill a graph globally for passage re-ranking scenario from an existing knowledge graph by pruning some unreliable or noisy relations based on TransE embedding. Then for a specific query-passage pair, we extract entities from both the query and passage, and construct a query-document bipartite entity graph based on query and passage entities and their k-hop neighbors, namely knowledge meta graph. **Challenge 1.** could be addressed in this distillation process.

For *knowledge aggregation*, we design a novel interaction module between text and knowledge graph to combine the implicit and explicit knowledge. To derive implicit knowledge from text, we employ PLM as text encoder. To be aligned with implicit knowledge, knowledge meta graph is encoded with a multi-layer graph neural network (i.e. k-hop), namely Graph Meta Network (GMN). Each transformer layer outputs word representations. Each graph meta network layer outputs entity representations. Both word and entity representations are aggregated as the input of the following transformer and GMN layer, respectively in a novelly designed module, namely knowledge injector. Therefore through knowledge aggregation, implicit knowledge from text corpus and explicit knowledge from existing knowledge graph can mutually boost each other to achieve a better re-ranking performance, in which the issues in **Challenge 2.** could be mitigated.

Overall, our contributions can be summarized as follows:

- It is the first attempt to solve the knowledge enhanced PLMs problem for passage re-ranking. The key motivation lies in that bridging the semantic gap between the query and passage with the help of both kinds of knowledge.
- We design a novel knowledge graph distillation method. It refines a reliable knowledge graph from the existing one globally and constructs a knowledge meta graph based on the refined graph locally.
- We propose a novel aggregation of PLM and graph neural network framework to model the interaction between explicit knowledge and implicit knowledge.
- Experimental results show the effectiveness of KERM on both general and domain specific data, achieving state-of-the-art performance for passage re-ranking. We also conduct a comprehensive study for the effects of each module in our method. The code is available at https://github.com/DQ0408/KERM.

2 RELATED WORK

In this section, we introduce several recently-proposed PLMs based re-rankers and retrievers. Moreover, we also present the general background of the related techniques involved in this paper, i.e. Knowledge Enhanced Pre-trained Language Models (KE-PLMs) and Graph Neural Network.

2.1 PLMs based Re-rankers

Existing PLMs based re-rankers typically improve ranking performance from two aspects: (1) **By optimizing the ranking procedure**: monoBERT [31] is the first work that re-purposed BERT as a passage re-ranker and achieves state-of-the-art results. duoBERT [32] integrates monoBERT in a multistage ranking architecture and adopts a pairwise classification approach to passage relevance computation. UED [47] proposes a cascade pre-training manner that can jointly enhance the retrieval stage through passage expansion with a pre-trained query generator and thus elevate the re-ranking stage with a pre-trained transformer encoder. The two stages can facilitate each other in a unified pre-training framework. H-ERNIE [2] proposes a multi-granularity PLM for web search. (2) **By designing rational distillation procedure**: LM Distill + Fine-Tuning [10] explores a variety of distillation methods to equip a smaller re-ranker with both general-purpose language modeling knowledge learned in pre-training and $search\text{-}specific$ relevance modeling knowledge learned in fine-tuning, and produces a faster re-ranker with better ranking performance. CAKD [14] proposes a cross-architecture knowledge distillation procedure with a Margin-MSE loss, which can distill knowledge from multiple teachers at the same time. RocketQAv1 [34] trains dual-encoder and cross-encoder in a cascade manner, which leverages the powerful cross-encoder to empower the dual-encoder. RocketQAv2 [36] proposes a novel approach that jointly trains the dense passage retriever and passage re-ranker. The parameters of RocketQAv2 are inherited from RocketQAv1. Besides, RocketQAv2 utilizes a large PLM for data augmentation and denoising, which can also be regarded as a distillation procedure. Notably, these two types of studies anticipate more insightful information to be captured by the advanced ranking and training procedures, while neglecting the limitations of implicit knowledge extracted from noisy and heterogeneous data. Therefore, in this paper, we proposed the first knowledge-enhanced PLM based re-ranker, which thoughtfully leverages explicit external knowledge that improve the effectiveness of the model.

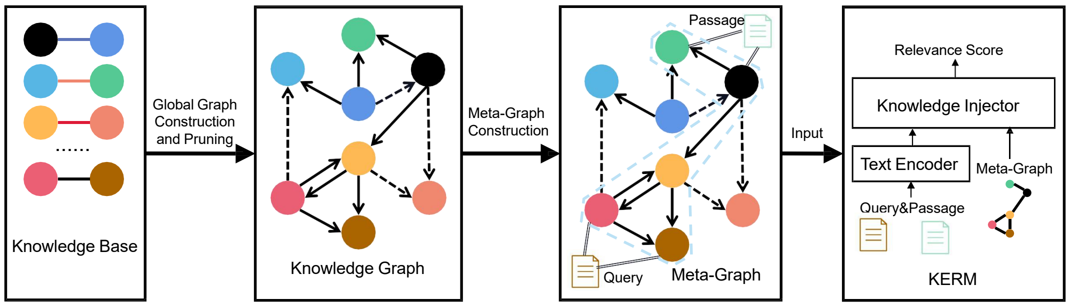

Figure 1: The workflow of KERM.

2.2 PLMs based Retrievers

The low-dimensional dense representations for query and passage are computed by PLMs based retrievers from the dual-encoder architecture. Afterward, the candidate passage set could be retrieved efficiently via approximate nearest neighbor algorithms. Existing studies could be categorized into two parts: (1) By optimizing the matching stage: DPR [16] is the first study to leverage PLM to empower the retriever by a single vector. Other researches, such as RepBERT [50], ColBERT [17], COIL [11] and Interactor [49], obtain multiple vectors for query and passage for matching. (2) By optimizing the representation learning module: RocketQAv1 [34] and RocketQAv2 [36] boost the representation learning of retriever by leveraging the power of cross-encoder in a cascade or joint manner. Other studies boost the representation learning by designed IR-oriented pre-training tasks. ICT [21] treats sentences as pseudo-queries and matched them to the passage they originate from. Condenser [9] utilizes a novel pre-training task, which can produces an information-rich representation to condense an input sequence.

2.3 Knowledge Enhanced Pre-trained Language Models (KE-PLMs)

Existing KE-PLMs can be categorized by the granularity of knowledge they incorporate from knowledge graph (KG), as text-based knowledge, entity knowledge and KG meta-graphs. To integrate text-based knowledge, RAG [22] and KIF [7] first retrieve top-k documents from Wikipedia using KNN-based retrieval, and the PLM model is employed to generate the output conditioned on these retrieved documents. Entity-level information can be highly useful for a variety of natural language understanding tasks. Hence, many existing KE-PLMs target this type of simple yet powerful knowledge. ERNIE(BAIDU) [42] introduces a new pre-training strategy of language model which masking phrases or entities in order to implicitly learn both synaptic and semantic knowledge from these units. ERNIE(THU) [51] integrates informative entity representations in the knowledge module into the underlying layers of the semantic module based on the alignments between text and entity to equip the model with the ability of knowledge awareness. As knowledge graphs provide richer information than simply entity, more and more researchers start to explore integration of more sophisticated knowledge, such as meta-graphs in KG. CokeBERT [40] proposes a novel semantic-driven Graph Neural Network (GNN) to dynamically select contextual knowledge and embed knowledge context according to textual context for PLMs, which can avoid the effect of redundant and ambiguous knowledge in KGs that cannot match the input text. CoLake [41] also uses GNN to aggregate information from the constructed meta-graph in both pre-training and inference. CoLake converts the meta-graph into token sequence and appends it to input sequence for PLMs, which is distinctive to CokeBERT. Although extensive research has been proposed up to now to address the knowledge-aware problem, none exists which constrained on how to use knowledge to empower PLMs particularly for re-ranking tasks.

2.4 Graph Neural Network

Existing Graph Neural Networks (GNNs) mainly fall into two categories: graph-based and path-based. Graph-based GNNs learn the structured information by directly passing nodes massage on the graph structure. GCNs [20] introduce a novel approach on graph-structured data by aggregating messages from its direct neighbors to learn the graph-structured feature efficiently and effectively. R-GCNs [38] are developed specifically to encode the highly multi-relational graphs by defining relation-specific weight matrix for each edge type. In contrast, path-based GNNs first decompose the graph into paths and then pass nodes massage on the path level, which can naturally utilize the relationship between neighbors to transmit messages. RNs [37] use MLPs to encode all paths in a graph and then pool the representation of paths to generate a global representation for the graph. KagNet [24] is a combination of GCNs, LSTMs and a hierarchical path-based attention mechanism, which forms an architecture for modeling nondegenerate paths in a graph. In this work, we use path-based GNNs to formulate our GMN module for its good scalability on modeling relationship information in heterogeneous graphs.

3 PROBLEM FORMULATION

3.1 Passage Re-ranking

Given a query $\mathbf{q}$, passage re-ranking aims at ordering a set of $\varkappa$ passages, i.e., $\mathcal{P} = \left\{\mathbf{p}_\kappa\right\}_{\kappa=1}^{\varkappa}$, which is usually retrieved from a large-scale passage collection by a retriever, e.g. BM25 [48], DPR [16] etc. In particular, a passage is a sequence of words $\mathbf{p} = \{w_p\}_{p=1}^{|\mathbf{p}|}$, where $|\mathbf{p}|$ is the length of passage $\mathbf{p}$. Similarly, a query is a sequence of words $\mathbf{q} = \{w_q\}_{q=1}^{|\mathbf{q}|}$. Note that a passage $\mathbf{p}$ consists of T sentences $\mathbf{p} = \{\mathbf{s}_\tau\}_{\tau=1}^{T}$.

Following a previous study [52], a desirable re-ranker is a scoring function $f^*(\cdot,\cdot)$ that maximizes the consistency between its predictions (denoted as $\hat{Y}_{\mathbf{q},\mathcal{P}} = \{f(\mathbf{q},\mathbf{p}_\kappa) \mid \mathbf{p}_\kappa \in \mathcal{P}\}$) and the ground truth labels (denoted as $Y = \{y_\kappa\}_{\kappa=1}^{\varkappa}$), i.e.,

$$f^* = \max_f \mathbb{E}_{\{\mathbf{q},\mathcal{P},Y\}} \vartheta(Y, \hat{Y}_{\mathbf{q},\mathcal{P}}), \tag{1}$$

where ϑ is a ranking metric (e.g., MRR@10) that measures the consistency between the predictions and the labels.

3.2 Explicit Knowledge Enhanced Passage Re-ranking

A knowledge base is usually represented as a directed graph $\mathcal{G} = \{\mathcal{E},\mathcal{R}\}$, where the node set $\mathcal{E}$ represents entities, and the edge set $\mathcal{R}$ is composed of relations between entities. A triplet (e_h, r, e_t) is the basic unit in the knowledge graph, where $e_h, e_t \in \mathcal{E}$ are head and tail entity respectively, and $r \in \mathcal{R}$ refers to their relations. For example, $(apple, used\_for, eating)$ means that "apple is used for eating".

To leverage explicit knowledge in $\mathcal{G}$ for passage re-ranking, we anticipate building a novel knowledge-enhanced passage re-ranker, whose objective can be defined as

$$f^* = \max_f \mathbb{E}_{\{\mathbf{q},\mathcal{P},Y\}} \vartheta(Y, \hat{Y}_{\mathbf{q},\mathcal{P},\mathcal{G}}), \tag{2}$$

where $\hat{Y}_{\mathbf{q},\mathcal{P},\mathcal{G}} = \{f(\mathbf{q},\mathbf{p}_\kappa \mid \mathcal{G}) \mid \mathbf{p}_\kappa \in \mathcal{P}\}$, and $f(\mathbf{q},\mathbf{p}_\kappa \mid \mathcal{G})$ represents the ranking score that is aware of the explicit knowledge extracted from $\mathcal{G}$.

4 KERM

In this section, we introduce **K**nowledge **E**nhanced **R**e-ranking **M**odel (**KERM**), which leverages explicit knowledge that improves conventional cross-encoder for passage re-ranking. Notably, the main challenges of incorporating explicit knowledge are to 1) distill a knowledge graph that is useful for re-ranking task, and 2) aggregate the explicit knowledge with the current implicit knowledge in an appropriate manner that can improve the overall performance. Hence our proposed approach is mainly composed of two parts, i.e., **knowledge graph distillation** and **knowledge aggregation**, to tackle two challenges respectively. In the rest of this section, we first describe how to distill a reliable knowledge graph globally and build a knowledge meta graph locally from it for a specific query-passage pair. Then, we present how to combine the distilled knowledge graph and existing text corpus to derive a knowledge enhanced passage re-ranker.

4.1 Knowledge Graph Distillation

Existing knowledge graphs are usually incomplete and noisy. It is unsuitable for direct introduction of them to the current model. Specially, there is no knowledge base particularly for passage re-ranking task. For example, *ConceptNet* [39] is a general knowledge graph that contains common sense knowledge, where the information might not be useful for our passage re-ranking task. Therefore, it is critical for us to propose a knowledge graph distillation process from both global and local perspectives.

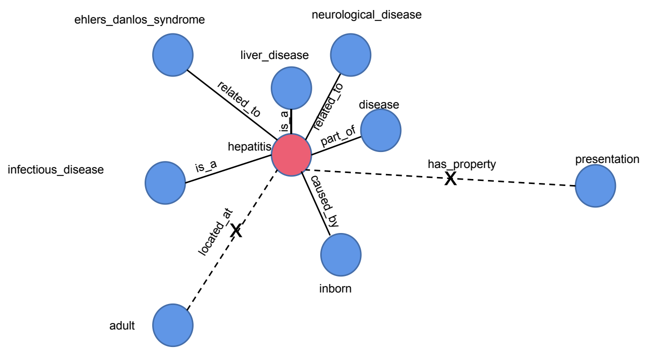

Figure 2: The illustration of global graph pruning.

4.1.1 Step 1: Global Graph Pruning.

Given a global knowledge graph $\mathcal{G}$, the first step is to eliminate those knowledge that might be noisy to be applied. To achieve this, we use TransE [1] to measure the reliability of a given knowledge triplet. In particular, TransE is an unsupervised learning method that learns latent representations for a knowledge triplet (e_h, r, e_t). Intuitively, it models the latent distribution of knowledge in a given knowledge graph, and those who are out of this distribution can be viewed as less informative knowledge, which should not be used. Based on this, we use the entity embeddings pre-trained by TransE to calculate a distance metric between two linked entities as

$$Rel_e(e_h, r, e_t) = \mathbf{E}(e_h)\cdot\mathbf{E}(r) + \mathbf{E}(e_h)\cdot\mathbf{E}(e_t) + \mathbf{E}(r)\cdot\mathbf{E}(e_t), \tag{3}$$

$$Dist(e_h, e_t) = \frac{1}{Rel_e(e_h, r, e_t)}, \tag{4}$$

where $\mathbf{E}(e)$ and $\mathbf{E}(r)$ are the TransE embeddings of entity and relation, respectively, and the inner product measures the relevance between two vectors. As the objective of TranE is aligned with minimizing the distance shown in Eq.(4), we can consider those knowledge triplets with small distance values as informative knowledge.

After measuring the reliability of knowledge, we prune $\mathcal{G}$ by only keep the top-Π neighboring entities $\mathcal{N}(e_h)$ of a given entity e_h, which can formally be defined as

$$\mathcal{N}(e_h) = \cup_{\pi=1}^{\Pi}\{e_t^{\pi}\}, where\, Dist(e_h, e_t^{\pi}) \le Dist(e_h, e_t^{\pi+1}). \tag{5}$$

Thus, the pruned global graph $\overline{\mathcal{G}}$ can be denoted as

$$\overline{\mathcal{G}} = \{(e_h, r, e_t) | e_h, e_t \in \mathcal{E} \wedge r \in \mathcal{R} \wedge e_t \in \mathcal{N}(e_h)\}. \tag{6}$$

Fig. 2 shows a real case of our global graph pruning method on ConceptNet, i.e., a general knowledge graph. In this case, the entity *hepatitis* has various relations to *disease*, *infectious disease*, *adult*, etc. From the distance of nodes in Fig. 2, we can clearly observe that the knowledge *hepatitis is an infectious disease* is more reliable and informative than *hepatitis is located at adult.* To *hepatitis*, the concept *adult* is more general than *infectious disease.* This indicates that our pruning method can effectively eliminate less informative knowledge.

4.1.2 *Step2: Meta-Graph Construction.*

Different from existing knowledge-enhanced PLMs for other NLP tasks, our aim for the re-ranking task is particularly on the relevance modeling between query and passage. Thus, we further leverage the knowledge in the global graph $\overline{\mathcal{G}}$ to construct "bridges" between query and passage, which alleviates the semantic gap and improves semantic modeling. More specifically, for a given query-passage pair (i.e., $(\mathbf{q}, \mathbf{p})$), we propose to construct a bipartite *meta-graph* that connects those entities in the $\mathbf{q}$ and those in $\mathbf{p}$.

Algorithm 1: Meta-Graph Construction Algorithm

```
Input: query $\mathbf{q}$, passage $\mathbf{p}$ and pruned global graph $\overline{\mathcal{G}}$
Output: meta-graph $\mathbf{G}$ of query $\mathbf{q}$ and passage $\mathbf{p}$

% key sentence selection;
$\mathbf{s}^* = \arg\max_{\mathbf{s}_i} Rel_{qs}(\mathbf{q}, \mathbf{s}_i)$ as defined in Eq.(7);
% target entity recognition;
EntityRecognition($text$) begin
    $\phi = \{\}$;
    for $i \in [1..|text|]$ do
        for $length \in [max\_phrase\_length..1]$ do
            if $text[i : i + length] \in \mathcal{E}$ then
                $\phi = \phi \cup \{text[i : i + length]\}$;
                break;
            end
        end
    end
    return $\phi$;
end
$\phi_{\mathbf{q}}$ = EntityRecognition($\mathbf{q}$),$\phi_{\mathbf{s}^*}$ = EntityRecognition($\mathbf{s}^*$);
% meta-graph construction via path discovery;
k=1;$\mathbf{G} = \{\}$;queue = {};
for $e_h \in \phi_q$ do
    queue = queue $\cup \{(e_h,)\}$
end
while $k \leq K$ do
    nextHop = {};
    for $path \in$ queue do
        $e_h = path[-1]$;
        for $e_t \in \mathcal{N}(e_h)$ do
            if $e_t \in \phi_{\mathbf{s}^*}$ then
                $\mathbf{G} = \mathbf{G} \cup \{path + (r, e_t)\}$;
            else
                nextHop = nextHop $\cup \{path + (r, e_t)\}$;
            end
        end
    end
    k+=1;queue = nextHop;
end
;
```

The construction process is shown in Alg. 1, which contains three sub-steps: key sentence selection, target entity recognition and path discovery.

(1) **Key sentence selection**. The actual information need of a user usually concentrates on a small part of a relevant passage [12]. To this end, we mimic human judgment and only focus on the sentence of each passage that is the most related to a query [52]. In particular, we define the relevance score between a query $\mathbf{q}$ and a sentence $\mathbf{s}_i$ as

$$Rel_{qs}(\mathbf{q}, \mathbf{s}_i) = \frac{\sum_{q=1}^{|\mathbf{q}|} \mathbf{E}(w_q)}{|\mathbf{q}|} \cdot \frac{\sum_{s=1}^{|\mathbf{s}_i|} \mathbf{E}(w_s)}{|\mathbf{s}_i|}. \tag{7}$$

For the sake of efficiency, we initialize $\mathbf{E}(w)$ from Word2Vec [29] embedding. Based on Eq.(7), we select the most relevant sentence $\mathbf{s}^*$ in $\mathbf{p}$ to build the meta-graph for $\mathbf{q}$ and $\mathbf{p}$.

(2) **Target entity recognition**. Next, we select the entities in $\mathbf{q}$ and $\mathbf{s}^*$ to construct the meta-graph. Specifically, we only consider the entities that exactly match in $\mathcal{E}$. Meanwhile, we omit those entity phrases that are sub-sequences of other recognized entities. For example, in the query "what causes low liver enzymes", both "liver" and "liver enzyme" are entities, but the entity "liver enzyme" is more informative to be recognized as the target entity, and "liver" should be omitted.

(3) **Path discovery**. Finally, given the target entities of $\mathbf{q}$ and $\mathbf{s}^*$ (denoted as $\phi_{\mathbf{q}}$ and $\phi_{\mathbf{s}^*}$, respectively), we perform Breadth First Search (BFS) on $\overline{\mathcal{G}}$ to discover the paths within K-hop between $\phi_{\mathbf{q}}$ and $\phi_{\mathbf{s}^*}$. Note that we only keep the within-K-hop paths that might be the most useful for the downstream re-ranking task. Meanwhile, the knowledge could be complemented from the K-hop paths.

After taking the series of processes, the meta-graph $\mathbf{G}_{\mathbf{q},\mathbf{p}} = \{\mathcal{E}_{\mathbf{q},\mathbf{p}}, \mathcal{R}_{\mathbf{q},\mathbf{p}}\}$ is constructed with the multi-hop paths discovered between $\phi_{\mathbf{q}}$ and $\phi_{\mathbf{s}^*}$. Fig. 3 shows an example of the meta-graph, which contains rich knowledge about the semantic relevance between the query and passage. Notably, a better key sentence selector or entity linker such as Sentence-BERT [35] and DER [46] may benefit the

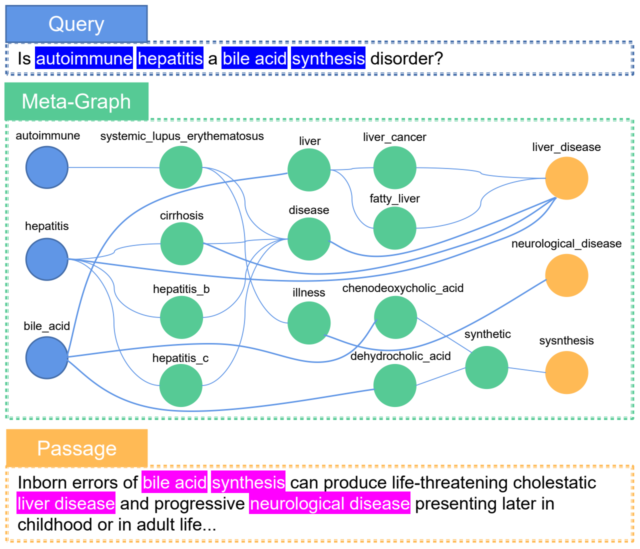
Query
Is autoimmune hepatitis a bile acid synthesis disorder?
Meta-Graph
autoimmune
systemic_lupus_erythematosus
liver
liver_cancer
liver_disease
hepatitis
cirrhosis
disease
fatty_liver
neurological_disease
bile_acid
hepatitis_b
illness
chenodeoxycholic_acid
synthetic
sysnthesis
hepatitis_c
dehydrocholic_acid
Passage
Inborn errors of bile acid synthesis can produce life-threatening cholestatic liver disease and progressive neurological disease presenting later in childhood or in adult life...

Figure 3: The illustration of a meta-graph constructed between the target entities of a query-passage pair. Note that some entities do not appear in the meta-graph, as they might not have any path that reaches the entities on the other side.

ranking performance, but can burden the entire model inference time, which is infeasible to a qualified re-ranker.

4.2 Knowledge Aggregation

Given a meta-graph $\mathbf{G_{q,p}}$, we propose a PLM based re-ranker that performs knowledge-enhanced relevance computation, i.e., $f(\mathbf{q},\mathbf{p} \mid \mathcal{G})$. In the following, we first introduce the text encoder, and then present how we inject explicit knowledge from $\mathbf{G_{q,p}}$ into the encoder.

4.2.1 *Text Encoder.*

We adopt the commonly-used cross-encoder as the text encoder. The input is formulated as the concatenation of a query-passage pair and the input layer converts the token indexes to a set of token embeddings [44] (i.e., $\mathbf{O}_0$) as

$$\mathbf{O}_0 = \text{InputLayer}([[CLS], \{w_q\}_{q=1}^{|\mathbf{q}|}, [SEP], \{w_p\}_{p=1}^{|\mathbf{p}|}, [SEP]]). \quad (8)$$

In the l-th transformer layer, text context features are extracted via multi-head self-attention and Feed Forward Network (FFN) as

$$\hat{\mathbf{H}}_l = \text{MultiHeadAttention}(\mathbf{O}_{l-1}), \quad (9)$$

$$\mathbf{O}_l = \sigma\left(\hat{\mathbf{H}}_l \mathbf{W}_l^1 + b_l^1\right)\mathbf{W}_l^2 + b_l^2, \quad (10)$$

where $\mathbf{W}_l$ and b_l are the parameters of FFN and σ is an activation function, and $\mathbf{O}_l$ is the output of layer l.

4.2.2 *Knowledge Injector.*

Based on the text encoder, we develop a knowledge injector that can seamlessly integrate explicit knowledge. Moreover, inspired by CokeBERT [40], our knowledge injector is equipped with a GMN module to dynamically refine the knowledge context on the basis of text context features learned by text encoder, which further improves the flexibility and usability of the knowledge enhancement. Besides, our method allows the text context and knowledge context to interact and mutually boost each other.

Knowledge injection. As shown in Fig. 4, the knowledge injector consists of multiple transformer layers, which is the same as the text encoder. Given a query-passage pair $(\mathbf{q},\mathbf{p})$, we first find the entities in $\mathbf{G_{q,p}}$ that can be enhanced by external knowledge. For these entities, we define $\mathbf{E}$ as the knowledge embeddings to be applied in the knowledge injection layers, where $\mathbf{E}$ is initialized by TransE embeddings extracted from the pruned global graph $\overline{\mathcal{G}}$.

Next, we align each entity with the first token of the corresponding phrase in the selected key sentence [51], and define the knowledge injection process as

$$\hat{\mathbf{H}}_l = \text{MultiHeadAttention}(\mathbf{O}_{l-1}), \quad (11)$$

$$\mathbf{F}_l = \sigma\left((\hat{\mathbf{H}}_l \mathbf{W}_l^1 + b_l^1) \oplus \Lambda(\mathbf{E}\mathbf{W}_l^3 + b_l^3)\right), \quad (12)$$

$$\mathbf{O}_l = \mathbf{F}_l \mathbf{W}_l^2 + b_l^2. \quad (13)$$

In Eq. (12), $\oplus$ means element-wise addition and $\Lambda(\cdot)$ represents the alignment function maps the entities to the corresponding positions of the tokens. By doing this, the external knowledge $\mathbf{E}$ is integrated in the output $\mathbf{O}_l$ of the knowledge injection layer. The final relevance score of this query-passage pair is defined as

$$f(\mathbf{q},\mathbf{p} \mid \mathcal{G}) = \sigma\left(\mathbf{O}_{\text{M}}^{[\text{CLS}]}\mathbf{W}^4 + b^4\right). \quad (14)$$

Knowledge propagation via meta-graph. It is worth noting that, the above-defined knowledge injection process only leverages knowledge embeddings learned by TransE on the global graph $\overline{\mathcal{G}}$. Particularly, it lacks considering the knowledge that bridges the semantics between query and passage. To this end, we introduce a Graph Meta Network (GMN) module that refines knowledge with the constructed meta-graph $\mathbf{G_{q,p}}$, The multi-hop paths of $\mathbf{G_{q,p}}$ allow the knowledge to be propagated between query and passage, which can enhance the relevance signal to be captured by the model, and thus alleviate the semantic gap.

More specifically, each knowledge injection layer has a multi-layer GMN (as shown in Fig. 4) to propagate knowledge on $\mathbf{G_{q,p}}$. First, the input of GMN is formulated with the fused feature $\mathbf{F}_l$ as

$$\hat{\mathbf{E}}_l^{(0)} = \Gamma(\mathbf{F}_l \mathbf{W}_l^5 + b_l^5), \quad (15)$$

where Γ represents the slice operation that extracts the fused information of the target entities in $\mathbf{G_{q,p}} = \{\mathcal{E}_{\mathbf{q,p}}, \mathcal{R}_{\mathbf{q,p}}\}$, and thus $\hat{\mathbf{E}}_l^{(0)}$ consists of fused entities representation $\hat{\mathbf{E}}_{e_1}^{(0)}, \hat{\mathbf{E}}_{e_2}^{(0)}, ..., \hat{\mathbf{E}}_{e_\Psi}^{(0)}$, i.e., $\Psi = |\mathcal{E}_{\mathbf{q,p}}|$.

Next, in the k-th layer of GMN, an entity embedding e_h is updated via an attentive aggregation from its neighbors $\mathcal{N}(e_h)$ as

$$\hat{\mathbf{E}}_{e_h}^{(k)} = \hat{\mathbf{E}}_{e_h}^{(k-1)} + \sum_{e_t \in \mathcal{N}(e_h)} \mathbf{a}_{ht}^{(k)} \hat{\mathbf{E}}_{e_t}^{(k-1)}. \quad (16)$$

Here, $\mathbf{a}_{ht}^{(k)}$ is the attention value, which can be defined as

$$\mathbf{a}_{ht}^{(k)} = \frac{exp(\mathbf{m}_{ht}^{(k)})}{\sum_{e_n \in \mathcal{N}(e_h)} exp(\mathbf{m}_{hn}^{(k)})}, \quad (17)$$

and the logits $\mathbf{m}_{ht}^{(k)}$ is computed as

$$\begin{aligned}\mathbf{m}_{ht}^{(k)} = \sigma\Big(&\alpha\left(\hat{\mathbf{E}}_{e_h}^{(k-1)} \| \hat{\mathbf{E}}_{e_t}^{(k-1)}\right) + \beta\left(\hat{\mathbf{E}}_{e_h}^{(k-1)} \| \hat{\mathbf{E}}_{r_{ht}}^{(k-1)}\right) \\ &+ \gamma\left(\hat{\mathbf{E}}_{r_{ht}}^{(k-1)} \| \hat{\mathbf{E}}_{e_t}^{(k-1)}\right)\Big).\end{aligned} \quad (18)$$

In Eq. (18), the functions $\alpha(\cdot)$, $\beta(\cdot)$ and $\gamma(\cdot)$ are full-connected layers, and $\cdot\|\cdot$ represents concatenation operation.

By applying a K-layer GMN in each layer of the knowledge injector, the output entity representation $\hat{\mathbf{E}}_{e_h}^{(K)}$ can ensemble knowledge from all the K-hop neighbors. As described in Section 4.1.2 that all the paths of $\mathbf{G_{q,p}}$ between $\mathbf{q}$ and $\mathbf{p}$ is within K hops, the GMN module can attentively propagate knowledge along the paths from entities in $\mathbf{p}$ to those in $\mathbf{q}$, and vice versa, which can enrich the semantics of the entities that benefit the relevance modeling.

Subsequently, the updated entity embeddings could be used as the knowledge to be injected in the next layer, i.e., $\mathbf{E} := \hat{\mathbf{E}}^{(K)}$. In other words, we can re-define Eq. (12) as

$$\mathbf{F}_l = \sigma\left((\hat{\mathbf{H}}_l \mathbf{W}_l^1 + b_l^1) \oplus \Lambda(\mathbf{E}_l \mathbf{W}_l^3 + b_l^3)\right), \quad (19)$$

where $\mathbf{E}_l$ is defined as

$$\mathbf{E}_l = \begin{cases} \hat{\mathbf{E}}_{l-1}^{(K)}, & l \in [2, M] \\ \text{TransE embeddings.} & l = 1 \end{cases} \quad (20)$$

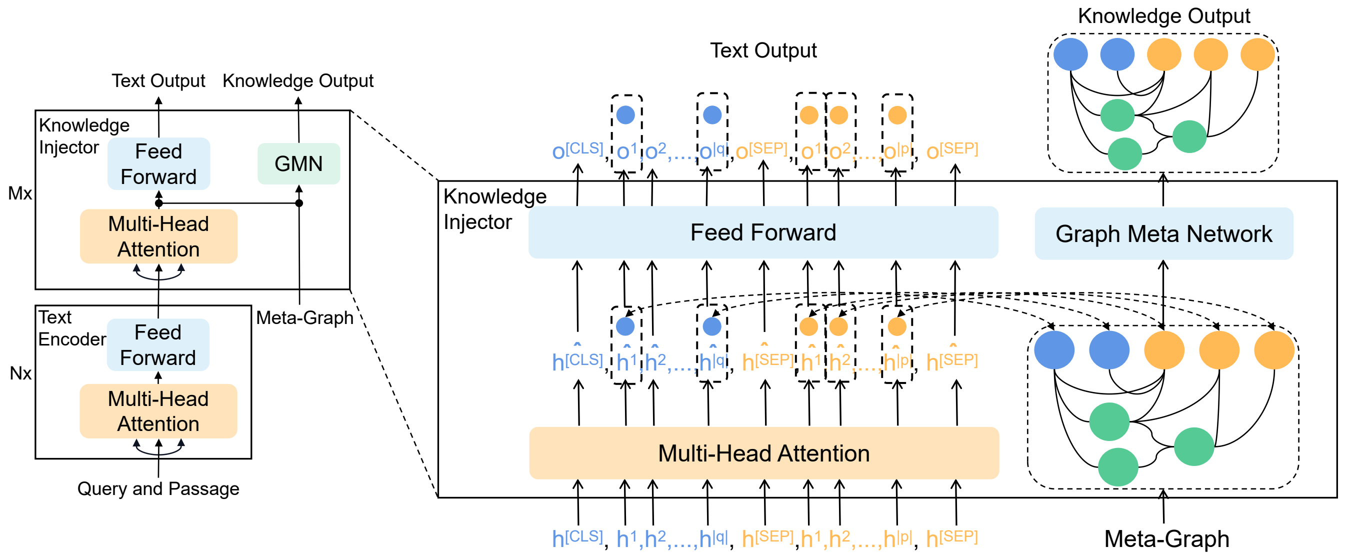

Figure 4: The architecture of KERM.

4.3 Model Optimization

Knowledge-enhanced pre-training. Following previous studies [18, 32, 47], we conduct continual pre-training on MSMARCO corpus to warm up the parameters of GMN module. We apply Masked Language Model (MLM) [4] and Sentence Relation Prediction (SRP) [45] as the pre-training tasks in KERM. Compared to conventional Next Sentence Prediction (NSP) [4], the task of SRP is to predict whether a given sentence is the next sentence, previous sentence relation or no relation with another sentence. To incorporate knowledge during the pre-training stage, we construct a meta-graph for each sentence pair, and apply the knowledge aggregation process as introduced above. The pre-training loss is defined as $\mathcal{L}_p = \mathcal{L}_{MLM} + \mathcal{L}_{SRP}$.

Knowledge-enhanced fine-tuning. We adopt a cross-entropy loss to fine-tune KERM:

$$\mathcal{L}_f = -\frac{1}{|Q|}\sum_{q\in Q}\log\frac{\exp(f(\mathbf{q},\mathbf{p}^+\mid\mathcal{G}))}{\exp(f(\mathbf{q},\mathbf{p}^+\mid\mathcal{G}))+\sum_{p^-}\exp(f(\mathbf{q},\mathbf{p}^-\mid\mathcal{G}))} \tag{21}$$

where $|Q|$ is the number of queries in training set, and p^+ and p^- denote the positive passage and negative passage in $\mathbb{P}$ for current query $\mathbf{q}$, respectively.

5 EXPERIMENTAL SETTING

5.1 Datasets

We use a large-scale public available corpus, i.e., MSMARCO-Passage collection [30], as our passage collection. This collection contains approximately 8.8 million passages extracted from 3.2 million web documents covering multiple fields. We train our model on the MSMARCO-TRAIN query set of 502,939 queries and evaluate KERM on three query sets. Table 1 provides the detailed information of these query sets. The first test set is **MSMARCO-DEV**, which includes 6,980 sparsely-judged queries mixed with multiple domains. Each query has an average of 1.1 relevant passages with binary relevance label. The second test set is **TREC 2019 DL** [3], which contains 43 densely-judged queries with fine-grained relevance labels, i.e., irrelevant, relevant, highly relevant and perfectly relevant. On average, a query has 95.4 relevant passages, and most queries have more than 10 relevant passages. With fine-grained labels and multiple relevant passages per query, TREC 2019 DL can be used to reflect the fine-grained ranking performance between relevant passages. To evaluate KERM on specific domains, we further introduce **Ohsumed** [1] query set, which contains 63 queries on bio-medical domain. The collection of Ohsumed is constructed from the first 20,000 passages in Mesh categories of the year 1991. Following the previous work [15], the test collection including 10,000 passages are utilized for performance comparison on Ohsumed query set. Each query has an average of 50.9 relevant passages with three graded relevance labels. In section 6.4, we demonstrate that the quality of external knowledge constructed by KERM in such domain could be more useful.

We use **ConceptNet** [39], a general knowledge graph as our external knowledge base $\mathcal{G}$. Following KagNet [24], we merge relation types to increase graph density and construct a multi-relational graph with 17 relation types, including *atlocation*, *causes*, *createdby*, etc.

5.2 Baselines

We include several PLMs based re-rankers in our evaluation, including the state-of-the-art:

- **monoBERT** [31]: The first study that re-purposes BERT as a re-ranker and achieves state-of-the-art results.
- **duoBERT** [32]: This work proposes a pairwise classification approach using BERT, which obtains the ability to be more sensitive to semantics through greater computation.

[1]http://disi.unitn.it/moschitti/corpora.htm

Table 1: Statistics of datasets.

	#Queries	#Rel.Psgs.	Rel.Psgs./Query	#Graded.Labels	Domain
MSMARCO-DEV	6980	7437	1.1	2	General
TREC 2019 DL	43	4102	95.4	4	General
OHSUMED	63	3205	50.9	3	Bio-Medical

- **UED** [47]: A unified pre-training framework that jointly refines re-ranker and query generator. For a fair comparison, we only use the re-ranker in UED without passage expansion.
- **LM Distill+Fine-Tuning (LDFT)** [10]: A variety of distillation methods are compared in this paper. The experimental results indicate that a proper distillation procedure (i.e. first distill the language model, and then fine-tune on the ranking task) could produce a faster re-ranker with better ranking performance.
- **CAKD** [14]: This work proposes a cross-architecture knowledge distillation procedure with Margin-MSE loss, which can distill knowledge from multiple teachers.
- **RocketQAv1** [34]: This work mainly focuses on the training of PLM based retriever, where the re-ranker is an intermediate product of its training process.
- **RocketQAv2** [36]: Based on RocketQAv1, this work proposes a novel approach that jointly trains the PLM based retriever and re-ranker.

To compare the performance of different methods, we resort to two ranking metrics. For MSMARCO-DEV, We adopt Mean Reciprocal Rank (i.e., MRR@10). For TREC 2019 DL, we use Mean Average Precision, i.e., MAP@10 and MAP@30. For Ohsumed, both Mean Reciprocal Rank and Mean Average Precision (i.e., MRR@10 and MAP@10) are employed for comprehensive performance analysis in queries requiring in-depth domain knowledge.

5.3 Implementation Details

We use the traditional sparse retriever BM25 [48] as our first stage method. All experiments are conducted under the **same BM25 setting** with 1000 retrieved candidates. We conduct experiments with the deep learning framework PaddlePaddle [27] on up to 4 NVIDIA Tesla A100 GPUs (with 40G RAM). For the GMN module, we use Paddle Graph Learning (PGL) [2], an efficient and flexible graph learning framework based on PaddlePaddle. For training, we used the Adam optimizer [19] with a learning rate of 1e-5 for text encoder and 1e-4 for knowledge injector. The model is trained up to 5 epochs with a batch size of 640 and 240 for base and large models respectively. In our experiments, the PLM small, base and large models have 6, 12 and 24 Transformer layers respectively. The text encoder has 9 layers and 21 layers for base and large model respectively, and the knowledge injector both has 3 layers in our experiment. The dropout rates are set to 0.1. The ratio of the positive to the hard negative is set to 1:19. All transformer layers in KERM's backbone are initialized from ERNIE-2.0 base [43], which is a BERT-like model pre-trained with a continual pre-training framework on multiple tasks. We perform Knowledge-enhanced pre-training on MARCO passage collection to warm up the parameters in knowledge injector, which has 60,000 iterations under the batch size of 256. For a fair comparison, the same pre-training without knowledge enhancement is also conducted on ERNIE$_{\text{base}}$ re-ranker and all models in ablation studies.

[2]https://github.com/PaddlePaddle/PGL

6 EXPERIMENTAL RESULTS

Here we compare ranking performances of KERM and other PLMs based re-rankers on the first two widely used query sets. Moreover, ablation studies for each component of KERM are also explored. All experimental results were reported under the same BM25 setting.

6.1 Effectiveness Analysis

Table 2 shows the ranking performance of KERM and baselines on MSMARCO-DEV and TREC 2019 DL. In the second column, model settings are displayed, including the PLMs used in models, whether distillation is enabled and computing resources required for model training. From Table 2, we observe the following phenomena.

(1) Compared with the best SOTA methods on the sparsely-judged MARCO-DEV query set, KERM outperforms all other baseline models except RocketQAv2. It utilizes a well-trained cross-encoder ERNIE$_{\text{large}}$ in RocketQAv1 to remove the predicted negatives with low confidence scores and include the predicted positives with high confidence scores. This can be regarded as a distillation. Meanwhile, RocketQAv2 achieves promising performance through a very large batch size on enormous computational resources, which is hardly comparable to our technique that only requires up to 4 GPUs. In addition to RocketQAv2, both KERM$_{\text{base}}$ and KERM$_{\text{large}}$ exceed strong baseline models, including duoBERT with sophisticated multiple re-ranking stages and CAKD distilled from multiple large models. It demonstrates the effectiveness of external knowledge injection.

(2) Among both kinds of baselines, KERM$_{\text{large}}$ achieves the best performance on the densely-judged TREC 2019 DL query set. MAP @10 and MAP@30 measure the quality of the ranking result over related passages. Baseline models with larger networks usually perform better in MAP, which indicates that complex structure helps model capture the fine-grained differences between related passages. With the well-designed GMN module and introduced reliable external knowledge, KERM$_{\text{base}}$ achieves the best performance on MAP@10 even compared to various large baseline models.

(3) Distilled models typically perform better at putting the relevant passage at top positions, but the subtle differences between relevant passages cannot be captured effectively through relatively small distilled models. On the MARCO-DEV query set, LDFT [10] performs better than duoBERT on MRR@10 and the former model size is much smaller than the latter. It shows that distillation plays a great role in performance improvement. Because LDFT [10] neither release the code nor report MAP in the original paper, we omit

Table 2: Performance comparison on MARCO-DEV and TREC 2019 DL.

	Model Settings			MARCO DEV Queries	TREC 2019 DL Queries	
	PLM	Distillation	Batch Size × GPU	MRR@10	MAP@10	MAP@30
BM25(anserini)	w/o	w/o	-	18.7	11.3	20.1
monoBERT	$\text{BERT}_{\text{large}}$	w/o	128×1	37.2	16.5	30.1
duoBERT[1]	$\text{BERT}_{\text{large}}$	w/o	128×1	39.0	16.8	30.9
UED[2]	$\text{BERT}_{\text{large}}$	w/o	64×4	39.5	-	-
LDFT	$\text{BERT}_{\text{small}}$	$\text{BERT}_{\text{base}}$	64×4	35.6	-	-
CAKD[3]	$\text{BERT}_{\text{small}}$	Multiple $\text{BERT}_{\text{large}}$	32×1	39.1	15.8	30.6
RocketQAv1	$\text{ERNIE}_{\text{large}}$	w/o	64×4	39.2	16.6	30.9
RocketQAv2	$\text{ERNIE}_{\text{base}}$	$\text{ERNIE}_{\text{large}}$	384×32	**40.1**	16.5	31.8
$\text{ERNIE}_{\text{base}}$	$\text{ERNIE}_{\text{base}}$	w/o	160×4	38.9	15.6	29.5
$\text{KERM}_{\text{base}}$	$\text{ERNIE}_{\text{base}}$+GMN	w/o	160×4	39.6	17.2	31.4
$\textbf{KERM}_{\text{large}}$	$\text{ERNIE}_{\text{large}}$+GMN	w/o	60×4	**40.1**	**17.8**	**33.2**

[1] duoBERT has one more re-ranking stage than the others.
[2] Only UED's re-ranker is used for report without passage expansion.
[3] CAKD uses a slightly better BM25 retriever on MARCO-DEV in its experiments.

its result on TREC 2019 DL query set. Additionally, models that perform well on MAP do not lead in MRR and vice versa, demonstrating that two metrics are to measure different aspects of the ranking quality. KERM shows the most stable performance on both metrics among all baseline models.

(4) Compared with $\text{ERNIE}_{\text{base}}$ we trained, $\text{KERM}_{\text{base}}$ shows a significant improvement on both two query sets. This indicates the explicit introduction of external knowledge can alleviate the semantic gap and heterogeneity between query and passage, and improve the semantic matching performance.

6.2 Ablation Study for Knowledge Injector

Knowledge injector module including knowledge injection and propagation process realized as Graph Meta Network (GMN), is mainly responsible for the interaction between text and knowledge graph. To explore their roles in the ranking performance, we remove the knowledge injection, aggregation process and the whole module separately and keep other units unchanged in KERM. Experimental results of three base models are shown in Table 3. KERM without knowledge injector module is degraded to vanilla ERNIE. KERM without knowledge propagation process is formally equivalent to ERNIE(THU) [51]. KERM without knowledge injection process takes the text of query-passage pair and meta graph as separate inputs, and then concatenate two parts of outputs to feed into a linear layer by redefining Eq.(19) and Eq.(14) respectively as

$$\mathbf{F}_l = \begin{cases} \sigma\left(\hat{\mathbf{H}}_l \mathbf{W}_l^1 + b_l^1\right), & \text{for Eq.(13)} \\ \sigma\left(\mathbf{E}_l \mathbf{W}_l^3 + b_l^3\right), & \text{for Eq.(15)} \end{cases} \tag{22}$$

$$f(\mathbf{q}, \mathbf{p} \mid \mathcal{G}) = \sigma\left(\left(\mathbf{O}_{\mathrm{M}}^{[\mathrm{CLS}]} \| \mathbf{E}_{\mathrm{M}}^{(K)}\right) \mathbf{W}^6 + b^6\right). \tag{23}$$

Table 3 shows the performance comparisons between different settings of knowledge injector, which is statistically significant. From this table, we can observe the following phenomena. (1) MRR@10 of KERM without interaction and propagation process decreases at least 1% respectively. This indicates both knowledge interaction and propagation processes play an indispensable role in ranking performance. (2) The performance of KERM without propagation is comparable to vanilla ERNIE. Not only query and passage entities, but also their multi-hop neighbors are essential for the ranking performance. (3) MRR@10 of KERM without knowledge interaction drops the most. It suggests the simple and straightforward way to aggregate knowledge graph with text does not work in the passage re-ranking scenario. The text and knowledge graph need to be refined with each other mutually in the interaction, which will be further analyzed in detail as follows.

Table 3: Performance Comparisons between different settings of knowledge injector on MSMARCO DEV.

	MRR@10
KERM	39.6
w/o Knowledge Interaction	38.5
w/o Knowledge Propagation	38.9
vanilla ERNIE	38.9

Table 4: MRR@10 results with different number of layers in knowledge injector on MSMARCO.

	#Layers					
	0	1	2	3	4	5
MRR@10	38.9	39.1	39.4	**39.6**	39.4	39.2

To further explore the text-knowledge interaction influence on the ranking performance, we compare ranking performances from KERM with different numbers of knowledge injector layers. All experiments in Table 4 are conducted with the same experimental settings except the number of knowledge injector layers (denoted as M). Note that in our setting, the number of text encoder layers N plus M is always 12, i.e. the number of layers in $\text{ERNIE}_{\text{base}}$. No knowledge injector layer ($M = 0$) represents the vanilla $\text{ERNIE}_{\text{base}}$ re-ranker without explicit knowledge enhancement.

With the increase of M in Table 4, the ranking performance is not improved linearly. Instead, the performance achieves the best at $M = 3$ and then falls down (statistically significant). This performance variation trend is contradictory to our intuition that the more injector layers, the deeper interaction between text and knowledge, and the more performance improvement is expected. The possible reason lies in that the knowledge injector layer makes pretrained parameters from ERNIE$_{base}$ not reusable, which means the implicit knowledge learned from large-scale is not applicable to these layers. Hence the number choice of the knowledge injector layer is somehow determined by the trade-off between implicit and explicit knowledge.

6.3 Ablation Study for Knowledge Graph Distillation

Knowledge graph distillation is performed in both global and local perspectives. To explore their roles in the ranking performance, we remove the graph pruning globally and sentence selection locally respectively, keep other settings unchanged, and derive KERM without graph pruning and sentence selection respectively. From results on TREC 2019 DL in Table 5, observations are listed as below. (1) Without global graph pruning, MRR@10 and the average edge score, calculated through Eq.(3), decrease the most, and the time efficiency drops slightly. This indicates the original knowledge graph exists noise data that affect performance. (2) Without sentence selection, the time efficiency drops the most and the average edge score decreases slightly, which proves that not every sentence in a passage has a positive effect on semantic matching. Overall, knowledge graph distillation is significant to KERM.

Table 5: Comparisons on different settings of knowledge graph distillation on TREC 2019 DL.

	Avg. Edge Score	MRR@10	Time
KERM	54.6	**98.4**	21.4ms
w/o Graph Pruning	51.4	96.1	27.0ms
w/o Sentence Selection	53.1	97.2	60.1ms

6.4 Quantitative Analysis on Different Domains

We further investigate the ranking effect of KERM on a specific domain. Specifically, we conduct experiments on OHSUMED from bio-medical field, and a bio-medical query subset of MSMARCO-DEV including 1, 110 queries. This query subset is derived from the mixed domain query set of MSMARCO-DEV by k-means clustering method [13], while the remaining subset with 5, 870 queires is denoted as the general domain subset. Performance comparisons between KERM and BM25, ERNIE are shown in Table 6.

Results are obtained from Table 6. (1) Poor ranking performances of all models on bio-medical domain indicates that it is more challenging in the data scarcity scenario, where textual data is not covered widely in the PLMs' pretraining datasets. (2) Compared with ERNIE, KERM has a higher relative improvement in bio-medical domain than general domain. This demonstrates that the incorporation of knowledge graph is more useful for a data scarcity domain. To verify this idea, we compare the size of knowledge meta graph used for different domains as follows.

We quantify the knowledge desirability as the size of average knowledge meta graph used in one domain. Specifically, we use the average number of edges as the size and average edge score calculated through Eq.(3) as the reliability of the knowledge meta graph. From Table 7, we can see that the meta-graph constructed on Bio-Medical domains is better in terms of quantity and quality. It indicates that the external knowledge found on professional domains contains more information.

Table 6: Ranking performance comparison on different domains.

	MARCO Dev Queries		Ohsumed Queries	
	General	Bio-Medical		
	MRR@10	MRR@10	MRR@10	MAP@10
BM25	19.2	16.4	69.6	9.5
ERNIE	38.5	30.5	79.7	10.1
KERM	39.7	32.1	81.2	11.0

Table 7: Quantitative analysis of knowledge in meta-graph on different domains.

	Avg. Edge NO.	Avg. Edge Score
MARCO General	47.9	50.9
MARCO Bio-Medical	57.7	53.9
Ohsumed	58.6	53.5

7 CONCLUSION

The main goal of this paper is to reasonably introduce external knowledge graph to PLMs for passage re-ranking. We first design a novel knowledge meta graph construction method to distill reliable and query related knowledge from a general and noisy knowledge graph. The knowledge meta graph bridges the semantic gap between each query and passage. Then we propose a knowledge injector layer for mutually updating text and knowledge representations, which transformers word to entity representations for graph meta network, vice versa. KERM is pretrained with Masked Language Model (MLM) Sentence Relation Prediction (SRP) [38] tasks, and fine-tuned with cross entropy loss function. Experimental results on public benchmark datasets show the effectiveness of KERM compared with several baselines. The role of each module in KERM is also comprehensively analyzed in this work.

8 FUTURE WORKS

Despite that the knowledge graph distillation in our method is empirically shown to be effective for the final performance, the implementation of graph pruning and meta-graph construction is still based on simple heuristics. A more promising way of formulating a useful meta-graph is to jointly learn a graph generator with the reranker in an end-to-end fashion, which enables more flexibility. Besides, it is currently infeasible to exploit the external knowledge in the retrieval stage, which needs to exhaustively build massive meta-graphs for a large scale of candidates. A further study could focus on how to use external knowledge in PLM based retriever.

ACKNOWLEDGMENTS

This research work was funded by the National Natural Science Foundation of China under Grant No.62072447.

REFERENCES

[1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).

[2] Xiaokai Chu, Jiashu Zhao, Lixin Zou, and Dawei Yin. 2022. H-ERNIE: A Multi-Granularity Pre-Trained Language Model for Web Search. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*.

[3] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820* (2020).

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[5] Qian Dong and Shuzi Niu. 2021. Latent Graph Recurrent Network for Document Ranking. In *International Conference on Database Systems for Advanced Applications*. Springer, 88–103.

[6] Qian Dong, Shuzi Niu, Tao Yuan, and Yucheng Li. 2022. Disentangled Graph Recurrent Network for Document Ranking. *Data Science and Engineering* 7, 1 (2022), 30–43.

[7] Angela Fan, Claire Gardent, Chloe Braud, and Antoine Bordes. 2020. Augmenting transformers with knn-based composite memory for dialogue. *arXiv preprint arXiv:2004.12744* (2020).

[8] Yixing Fan, Xiaohui Xie, Yinqiong Cai, Jia Chen, Xinyu Ma, Xiangsheng Li, Ruqing Zhang, Jiafeng Guo, and Yiqun Liu. 2021. Pre-training Methods in Information Retrieval. *arXiv preprint arXiv:2111.13853* (2021).

[9] Luyu Gao and Jamie Callan. 2021. Condenser: a Pre-training Architecture for Dense Retrieval. *arXiv preprint arXiv:2104.08253* (2021).

[10] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2020. Understanding BERT rankers under distillation. *arXiv preprint arXiv:2007.11088* (2020).

[11] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. COIL: Revisit exact lexical match in information retrieval with contextualized inverted list. *arXiv preprint arXiv:2104.07186* (2021).

[12] Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W Bruce Croft, and Xueqi Cheng. 2020. A deep look into neural ranking models for information retrieval. *Information Processing & Management* 57, 6 (2020), 102067.

[13] John A Hartigan and Manchek A Wong. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)* 28, 1 (1979), 100–108.

[14] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving efficient neural ranking models with cross-architecture knowledge distillation. *arXiv preprint arXiv:2010.02666* (2020).

[15] Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*. Springer, 137–142.

[16] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[17] Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 39–48.

[18] Meoungjun Kim and Youngjoong Ko. 2021. Self-supervised Fine-tuning for Efficient Passage Re-ranking. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3142–3146.

[19] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[20] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[21] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300* (2019).

[22] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401* (2020).

[23] Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2020. PARADE: Passage representation aggregation for document reranking. *arXiv preprint arXiv:2008.09093* (2020).

[24] Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. *arXiv preprint arXiv:1909.02151* (2019).

[25] Yiding Liu, Weixue Lu, Suqi Cheng, Daiting Shi, Shuaiqiang Wang, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained Language Model for Web-scale Retrieval in Baidu Search. *arXiv preprint arXiv:2106.03373* (2021).

[26] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[27] Yanjun Ma, Dianhai Yu, Tian Wu, and Haifeng Wang. 2019. PaddlePaddle: An open-source deep learning platform from industrial practice. *Frontiers of Data and Domputing* 1, 1 (2019), 105–115.

[28] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1101–1104.

[29] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *In NAACL 2013: Human language technologies*. 746–751.

[30] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

[31] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[32] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424* (2019).

[33] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375* (2019).

[34] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 5835–5847.

[35] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[36] Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. RocketQAv2: A Joint Training Method for Dense Passage Retrieval and Passage Re-ranking. *arXiv preprint arXiv:2110.07367* (2021).

[37] Adam Santoro, David Raposo, David GT Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. 2017. A simple neural network module for relational reasoning. *arXiv preprint arXiv:1706.01427* (2017).

[38] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*. Springer, 593–607.

[39] Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

[40] Yusheng Su, Xu Han, Zhengyan Zhang, Yankai Lin, Peng Li, Zhiyuan Liu, Jie Zhou, and Maosong Sun. 2021. Cokebert: Contextual knowledge selection and embedding towards enhanced pre-trained language models. *AI Open* 2 (2021), 127–134.

[41] Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. Colake: Contextualized language and knowledge embedding. *arXiv preprint arXiv:2010.00309* (2020).

[42] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223* (2019).

[43] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 8968–8975.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[45] Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Jiangnan Xia, Liwei Peng, and Luo Si. 2019. Structbert: incorporating language structures into pre-training for deep language understanding. *arXiv preprint arXiv:1908.04577* (2019).

[46] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Zero-shot Entity Linking with Dense Entity Retrieval. CoRR abs/1911.03814 (2019). *arXiv preprint arxiv:1911.03814* (2019).

[47] Ming Yan, Chenliang Li, Bin Bi, Wei Wang, and Songfang Huang. 2021. A Unified Pretraining Framework for Passage Ranking and Expansion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4555–4563.

[48] Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1253–1256.

[49] Wenwen Ye, Yiding Liu, Lixin Zou, Hengyi Cai, Suqi Cheng, Shuaiqiang Wang, and Dawei Yin. 2022. Fast Semantic Matching via Flexible Contextualized Interaction. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1275–1283.

[50] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. Repbert: Contextualized text embeddings for first-stage retrieval. *arXiv preprint arXiv:2006.15498* (2020).

[51] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129* (2019).

[52] Lixin Zou, Shengqiang Zhang, Hengyi Cai, Dehong Ma, Suqi Cheng, Shuaiqiang Wang, Daiting Shi, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained language model based ranking in Baidu search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 4014–4022.