Towards Robust Ranker for Text Retrieval

Yucheng Zhou[1*], **Tao Shen**[1], **Xiubo Geng**[2], **Chongyang Tao**[2], **Can Xu**[2], **Guodong Long**[1], **Binxing Jiao**[2], **Daxin Jiang**[2†]
[1]Australian AI Institute, School of CS, FEIT, University of Technology Sydney
[2]Microsoft
yucheng.zhou-1@student.uts.edu.au, {tao.shen, guodong.long}@uts.edu.au
{xigeng,chongyang.tao,can.xu,binxjia,djiang}@microsoft.com

Abstract

A neural ranker plays an indispensable role in the de facto 'retrieval & rerank' pipeline, but its training still lags behind due to the weak negative mining during contrastive learning. Compared to retrievers boosted by self-adversarial (i.e., in-distribution) negative mining, the ranker's heavy structure suffers from query-document combinatorial explosions, so it can only resort to the negative sampled by the fast yet out-of-distribution retriever. Thereby, the moderate negatives compose ineffective contrastive learning samples, becoming the main barrier to learning a robust ranker. To alleviate this, we propose a multi-adversarial training strategy that leverages multiple retrievers as generators to challenge a ranker, where i) diverse hard negatives from a joint distribution are prone to fool the ranker for more effective adversarial learning and ii) involving extensive out-of-distribution label noises renders the ranker against each noise distribution, leading to more challenging and robust contrastive learning. To evaluate our robust ranker (dubbed R^{2}ANKER), we conduct experiments in various settings on the passage retrieval benchmarks, including BM25-reranking, full-ranking, retriever distillation, etc. The empirical results verify the new state-of-the-art effectiveness of our model.

1 Introduction

Text retrieval plays a crucial role in many applications, such as web search (Brickley et al., 2019) and recommendation (Zhang et al., 2019). Given a text query, it aims to retrieve all relevant documents from a large-scale collection[1] (Qu et al., 2021; Gao and Callan, 2022). For a better efficiency-effectiveness trade-off, the text retrieval *de facto* paradigm relies on a 'retrieval & rerank' pipeline (Guo et al., 2022). That is, 'retrieval' is to use a fast retriever to fetch a set of top document candidates given a query, while 'rerank' is to re-calculate the relevance of the query to each candidate by a heavy yet effective ranker for better results.

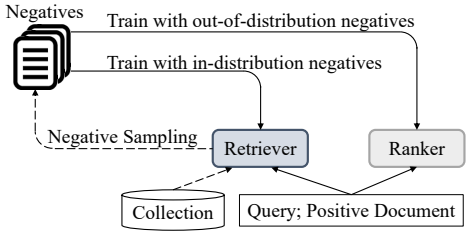


Figure 1: Retrieval models, the retriever and ranker, trained on in-distribution negatives and out-of-distribution negatives.

Differing from most natural language understanding (NLU) tasks defined as categorical classification (Zhang et al., 2015), training retrieval models, including the retriever and the ranker, are usually formulated as a contrastive learning problem. However, there are merely positive query-document pairs provided in most applications, regardless of negative samples. Hence, a critical prerequisite of the training is to sample negative documents from the collection for training queries.

As random sampling is prone to mine trivial negatives and proven less effective in training (a.k.a. in-batch negatives), a primary sampling method was proposed to leverage BM25 (Karpukhin et al., 2020) to fetch relatively challenging negatives for more effective training. In contrast to such an *out-of-distribution* negative sampling technique where the negatives mined by one retriever are used to train another, recent advanced negative mining methods resort to *in-distribution* sampling technique that leverages the retriever being trained to obtain the challenging-so-far negative documents from the collection. It has been proven in-distribution sampling is superior to out-of-distribution one as the former offers more model-specific contrastive learning samples towards the

[*]Work is done during internship at Microsoft.
[†]Corresponding author.
[1]while each collection entry could be a sentence, passage, document, etc., we adopt *document* for a clear demonstration.

Findings of the Association for Computational Linguistics: ACL 2023, pages 5387–5401
July 9-14, 2023

nuance among the positive and negatives for a given query (Qu et al., 2021; Ren et al., 2021b).

Nonetheless, a necessary condition of such an in-distribution sampling technique is the efficiency of the target model in large-scale retrieval. Unfortunately, compared to the bi-encoder based retriever that satisfies the efficiency requirement of large-scale retrieval, a cross-encoder based ranker suffers from combinatorial explosion brought by applying the heavy cross-encoder (e.g., the Transformer encoder) to every query-document concatenation. Thereby, the training of ranker can depend merely on out-of-distribution negative sampling technique – either by the BM25 retriever (Nogueira and Cho, 2019) or a trainable semantic retriever (Ren et al., 2021b) – leading to sub-optimal ranker due to a lack of adversarial training samples towards the expressively powerful cross-encoder.

In this paper, we aim to train a robust ranker by mining more challenging negatives and thus more effective contrastive samples. To this end, we propose a simple yet effective multi-adversarial training framework towards a robust ranker (R^2ANKER), where multiple retrievers as generators are integrated to mine diverse hard negatives and challenge a single ranker as the discriminator.

As such, R^2ANKER has certain merits regarding the robustness of its model training. First, intuitively, sampling negative over a joint distribution of various retrievers is more likely to offer more challenging hard negatives, which compensates the weakness of the previous single-retriever generator and makes the adversarial learning more robust. Second, as the false negatives are closely subject to the relevance distribution over the collection by a specific retriever, various negative generators achieved by different retrievers are prone to sample out-of-distribution or open-set label noise (Wei et al., 2021) to each other. In light of '*insufficient capacity*' assumption (Arpit et al., 2017), such open-set noise has been proven effective in improving robustness (Wei et al., 2021) when learning a ranker with open-set noises.

In experimentswe adopt several passage benchmark datasets (Nguyen et al., 2016) to evaluate our proposed model in various settings. Specifically, our method achieves new state-of-the-art performance on BM25 reranking and full-ranking on passage retrieval. Meantime, to verify the expressive power of our ranker model, we conduct an experiment to distill our well-trained model to a retriever, which shows state-of-the-art first-stage in terms of passage retrieval performance. Moreover, our extensive analyses unveil the essence regarding negative distributions to reach a robust ranker and also compare with negatives sampled by redistribution.

2 Related Work

Ranker for Information Retrieval. To achieve both efficiency and effectiveness, a de facto pipeline for large-scale retrieval is 'retrieval & rerank' (Guo et al., 2022). The 'retrieval' is to use a bi-encoder based retriever to encode queries and documents into dense representations and fetch out candidate documents relevant to the query through a lightweight metric (Gao and Callan, 2021). The 'rerank' aims to conduct a more accurate ranking on pairs of query and candidate documents by a cross-encoder based ranker (Ren et al., 2021b). Thereby, the ranker is a crucial part of the pipeline and directly affects the final performance of passage or document retrieval (Ren et al., 2021b; Zhou et al., 2022). In addition, rankers are currently widely used as a teacher in retriever training. The scores derived by the ranker are demonstrated that they can guide the retriever learning through knowledge distillation (Ren et al., 2021b; Zhang et al., 2022a). Moreover, rankers can be used to filter out top-retrieved documents that are likely to be false negatives (Qu et al., 2021). Therefore, the ranker not only directly affects the final performance of information retrieval but also can improve the performance of the retriever through knowledge distillation and false negative filtering. In this work, we propose a simple yet effective multi-adversarial training framework toward a robust ranker.

Ranker Training. Using negatives to train a ranker has proven effective in many works (Zhang et al., 2022a; Ren et al., 2021b). Since the ranker is based on the cross-encoder structure, a query and document need to be concatenated and passed to the ranker for relevance calculation. However, directly using the ranker to sample negatives on the collection suffers from combinatorial explosion. Therefore, many methods (Khattab and Zaharia, 2020; Qu et al., 2021) adopt the static hard negatives sampled from a retriever, which are fixed during ranker training. In addition, some methods (Zhang et al., 2022a; Ren et al., 2021b) introduce a joint training approach for dense passage retrieval and passage reranking, which dynamically update

both the parameters of the ranker and the retriever. Nevertheless, these methods can depend merely on out-of-distribution negative sampling techniques leading to sub-optimal rankers due to a lack of adversarial training samples towards the expressively powerful cross-encoder. Therefore, we integrate multiple retrievers regarded as generators to mine diverse hard negatives and challenge a single ranker as the discriminator.

Hard Negative Mining. Hard negative mining (Khattab and Zaharia, 2020; Zhang et al., 2022a; Qu et al., 2021) has been proven very effective in contrastive learning for text representation of retrievers. In contrast to random or in-batch negative sampling, it can find more challenging negatives for a pair of an anchor (i.e., query) and its positive example. They compose effective contrastive samples to help models learn against contextual nuance between the positive and negatives. At early stage, a large number of works employ the off-the-shelf BM25 retriever to fetch negative from a large collection (Karpukhin et al., 2020), which greatly boosts the retrievers. Furthermore, recent works (Gao and Callan, 2021, 2022) leverage a retriever to sample retriever-specific hard negatives for each query, which are considered the most challenging negatives. In this study, we sample negatives over a joint distribution of various retrievers, which is likely to offer more challenging hard negatives.

3 R^{2}anker: Robust Ranker

Task Formulation. Given a text query q, a ranker model, $\mathcal{K}(q,d)$, is responsible for calculating a relevance score between q and an arbitrary document d from a large-scale collection $\mathbb{D}$ (i.e., $d \in \mathbb{D}$). It usually serves as a downstream module for an efficient retriever, $\mathcal{R}$, to compose a 'retrieval & rerank' pipeline, where a lightweight retriever $\mathcal{R}$ (e.g., bi-encoder) is to retrieve top candidates and then a relatively heavy-structured $\mathcal{K}$ (says cross-encoder (Devlin et al., 2019)) to make the results better.

3.1 Contrastive Learning for Retrieval Model

Formally, the ranker $\mathcal{K}(q,d)$ is usually built upon a deep Transformer encoder for dense interactions in a pair of query and document (so called cross-encoder, or one-stream encoder), i.e.,

$$\mathcal{K}(q,d) \coloneqq s^{(\text{ce})} = \text{Transfm-Enc}(\texttt{[CLS]}q\texttt{[SEP]}d\texttt{[SEP]}; \theta^{(ce)}). \quad (1)$$

As each text query q must be concatenated with its every candidate document d to pass into the heavy Transformer encoder, it is impossible in terms of computation overheads to apply a ranker to large-scale retrieval (i.e., millions to billions of candidates). In contrast, a retriever $\mathcal{R}(q,d)$ is usually defined as a bi-encoder (a.k.a. dual-encoder, two-stream encoder, and Siamese encoder) to derive counterpart-agnostic representation vectors, i.e.,

$$\mathcal{R}(q,d) \coloneqq s^{(\text{bi})} = <\boldsymbol{u}, \boldsymbol{v}>, \text{ where,} \quad (2)$$
$$\boldsymbol{u} = \text{Transfm-Enc}(\texttt{[CLS]}q\texttt{[SEP]}; \theta^{(be)}),$$
$$\boldsymbol{v} = \text{Transfm-Enc}(\texttt{[CLS]}d\texttt{[SEP]}; \theta^{(be)}).$$

Here, $<\cdot,\cdot>$ denotes a lightweight relevance metric, e.g., dot-product and cosine similarity. As such, all the documents in $\mathbb{D}$ can be independently embedded and used for large-scale retrieval via the fast relevance metric.

Despite heterogeneous neural structures, training the retrieval models, i.e., the ranker in Eq.(1) and the retriever in Eq.(2), are both formulated as a contrastive learning problem. However, only positive document(s), d_+^q, is provided for each training query $q \in \mathcal{Q}^{(\text{trn})}$, regardless of its negative ones, i.e., $\mathbb{N}^q = \{d_-^q\}$, for contrastive learning. Note that if no confusion arises, we omit the subscript q indicating a specific q for clear writing. Therefore, to train a retrieval model, a prerequisite is determining a negative sampling strategy to make the training procedure more effective, i.e.,

$$\mathbb{N} = \{d | d \sim P(\mathbb{D} \backslash \{d_+\} | q; \theta^{(\text{smp})})\}, \quad (3)$$

where P denote a probability distribution over $\mathbb{D}$, which can be either non-parametric (i.e., $\theta^{(\text{smp})} = \emptyset$) or parametric (i.e., $\theta^{(\text{smp})} \neq \emptyset$).

Then, we take the ranker training for a demonstration: it calculates a probability distribution over $\{d_+\} \cup \mathbb{N}$, i.e.,

$$P(\text{d}|q, \{d_+\} \cup \mathbb{N}; \theta^{(\text{ce})}) = \frac{\exp(\mathcal{K}(q,d))}{\sum_{d' \in \{d_+\} \cup \mathbb{N}} \exp(\mathcal{K}(q,d'))}. \quad (4)$$

Lastly, the ranker is trained via a contrastive learning objective, whose training loss is defined as

$$L = -\sum\nolimits_{q,d_+} \log P(\text{d} = d_+ | q, \{d_+\} \cup \mathbb{N}; \theta^{(\text{ce})}). \quad (5)$$

3.2 Multi-Adversarial Ranker Training

A large amount of previous works (Qu et al., 2021; Ren et al., 2021b) have proven that the quality of

negative mining strategy significantly affects the performance of contrastive learning. As exhaustive training (i.e., $\mathbb{N} = \mathbb{D}\backslash\{d_+\}$) is infeasible in practice, how to train the model effectively with limited computation resources remains an open question.

Instead of random sampling, i.e., $\mathbb{N}^{(\text{rdm})} = \{d|d \sim \text{Uniform}(\mathbb{D}\backslash\{d_+\}|q)\}$, a recent trend is to leverage a retrieval model, especially a retriever $\mathcal{R}(\cdot,\cdot)$, to fetch the model-specific top-challenging negatives to train the retrieval model itself. This strategy is also known as self-adversarial training or hard negative mining (Zhang et al., 2022a; Qu et al., 2021). Formally, such a self-adversarial training technique to sample in-distribution negatives to train a retrieval model (i.e., $\theta^{(\text{smp})}$ in Eq.(3) equaling to $\theta^{(\text{ce})}$ of $\mathcal{K}$) can be written as

$$J = \max_{\theta^{(*)}} \mathbb{E}_{\mathbb{N}^{(*)}=\{d|d\sim P(\mathrm{d}|q,\mathbb{D}\backslash\{d_+\};\theta^{(*)})\}}[\log P(\mathrm{d}=d_+|q,\{d_+\}\cup\mathbb{N}^{(*)};\theta^{(*)})], \quad (6)$$

where $\theta^{(*)}$ parameterizes a retrieval model.

Despite efficacy, this self-adversarial technique cannot be applied to our targeted ranker training as it depends on the retrieval model's capability of large-scale retrieval, i.e., feasibility of calculating $P(\mathrm{d}|q,\mathbb{D}\backslash\{d_+\};\theta^{(*)})$ in Eq.(7) where $\mathbb{D}$ is huge. This is because $\mathcal{K}$ as a sampler over $P(\mathrm{d}|q,\mathbb{D}\backslash\{d_+\};\theta^{(\text{ce})})$ suffers from a combinatorial explosion problem brought from the cross-encoder, leading to intractable computation overheads. Practically, $\theta^{(\text{smp})}$ is must as efficient as possible to circumvent the problem, which could be a heuristic strategy (e.g., uniform sampling), lightweight term-based retriever (e.g., BM25), or later-interaction representation models (e.g., Siamese encoder).

As a remedy, the ranker training can resort to adversarial training (Zhang et al., 2022a), where an efficient retriever is used to sample top-hard out-of-distribution negatives for challenging the ranker. This can be formally written as

$$J^{\mathcal{R}^*,\mathcal{K}^*} = \min_{\theta^{(\text{be})}} \max_{\theta^{(\text{ce})}} \mathbb{E}_{\mathbb{N}^{(\text{be})}=\{d|d\sim P(\mathrm{d}|q,\mathbb{D}\backslash\{d_+\};\theta^{(\text{be})})\}}[\log P(\mathrm{d}=d_+|q,\{d_+\}\cup\mathbb{N}^{(\text{be})};\theta^{(\text{ce})})], \quad (7)$$

where the $\theta^{(\text{be})}$-parameterized $\mathcal{R}$ can be either a frozen and well-trained (Qu et al., 2021) or a jointly optimized (Zhang et al., 2022a) retriever.

Although learning from (adversarial) hard negatives has been proven effective to obtain a high-performing ranker (Ren et al., 2021b; Zhang et al., 2022a), a single retriever $\mathcal{R}$, even well-trained with various advanced techniques (Qu et al., 2021; Lu et al., 2022), is hard to provide hard enough negatives to challenge the ranker $\mathcal{R}$ for robust training.

Hence, we propose a multi-adversarial training strategy for ranker, where multiple heterogeneous retrievers are integrated to jointly sample negatives and challenge the only ranker for effective learning. As such, this ranker learning strategy is defined as

$$J^{\{\mathcal{R}_j^*\}_{j=1}^M,\mathcal{K}^*} = \min_{\Theta^{(\text{mul})}=\{\theta^{(\text{be})}\}_{j=1}^M} \max_{\theta^{(\text{ce})}} \mathbb{E}_{\mathbb{N}^{(\text{mul})}=\{d|d\sim P(\mathrm{d}|q,\mathbb{D}\backslash\{d_+\};\Theta^{(\text{mul})})\}}[\log P(\mathrm{d}=d_+|q,\{d_+\}\cup\mathbb{N}^{(\text{mul})};\theta^{(\text{ce})})], \quad (8)$$

where $\theta_j^{(\text{be})}$ parameterizes a retriever $\mathcal{R}_j$ (if applicable) and $\mathbb{N}^{(\text{mul})}$ is a set of negatives sampled by

$$P(\mathrm{d}|q,\mathbb{D}\backslash\{d_+\};\Theta^{(\text{mul})}) = \prod_{\theta_j^{(\text{be})}\in\Theta^{(\text{mul})}} P(\mathrm{d}|q,\mathbb{D}\backslash\{d_+\};\theta_j^{(\text{be})}). \quad (9)$$

Remark. Using heterogeneous retrievers can provide more diverse hard negatives – seen as different negatives distributions – prone to be more challenging for ranker training – leading to robust ranker. Here, the heterogeneity can assure by matching paradigm (term-based matching (Yang et al., 2017) v.s. semantic match (Gao and Callan, 2021)), representing paradigm (dense-vector embedding (Gao and Callan, 2022) v.s. lexicon-weighting embedding (Formal et al., 2021)), etc. By Zhang et al. (2022b), dense and lexical negatives are proven to provide more diverse views cf. BM25 negatives. As verified in experiments, the joint negative distribution is closer to the negative distribution under $\theta^{(\text{ce})}$, i.e., in-distribution $P(\mathrm{d}|q,\mathbb{D}\backslash\{d_+\};\theta^{(\text{ce})})$.

3.3 Robustness by Open-set Noise

Due to limited crowd-sourcing resources, it is impossible to exhaustively annotate the relevance of every query $\forall q \in \mathcal{Q}^{(\text{trn})}$ to every document $\forall d_+^q \in \mathbb{D}$. As such, sampling negatives by a strong retriever usually introduces a label-noise problem.

Fortunately, from the view of noise-labeling problem, we could employ the concept of 'open-set noise' to verify robustness of our ranker. As proven by a recent "*insufficient capacity*" (Arpit et al., 2017) assumption, learning a variety of out-of-distribution (OOD) or open-set noise can improve robustness against inherent label noises that are subject to one dataset or one distribution. Therefore, as multiple heterogeneous retrievers are involved in our multi-adversarial training framework to provide

mutually-OOD hard negatives, the ranker trained on these samples can be robust to the noises from every single retriever, resulting in a superior ranker after the training. Please see §A for more details.

3.4 Framework Grounding

Considering either diversity of hard negatives or distributions of open-set noises, the choice of heterogeneous retrievers in our multi-adversarial ranker training framework is vitally important. Based on the taxonomy of modern retrievers along with two axes, i.e., matching and representing paradigms, we opt in three representative retrievers: i) **BM25** retriever: A simple term-based BM25 retrieval model built on the whole collection; ii) **Den** retriever: A dense-vector semantic retriever, coCondenser (Gao and Callan, 2022); and iii) **Lex** retriever: A lexicon-weighing semantic retriever, SPLADE (Formal et al., 2021). Besides, we detail *retriever training* and *negatives sampling* in §B.

4 Experiments

We evaluate our ranker on following 3 retrieval tasks, and refer to §C for setups of *ranker training*, *retriever distillation*, and *retriever pre-training*.

BM25 Reranking Task. BM25 reranking uses a ranker to re-rank the top 1000 passages by BM25 for each query. Here, we adopt the popular passage retrieval datasets, MS-Marco (Nguyen et al., 2016) and TREC Deep Learning 2019 (Craswell et al., 2020), as well as their official BM25 retrieval candidates. Following previous work, we report the performance in MRR@10 on MS-Marco and NDCG@10 on TREC Deep Learning 2019.

Full Ranking Task. Full ranking leverages a ranker to rank the top-1000 passages retrieved from the full collection by a specific retriever. We adopt the MS-Marco dataset on which we will specify the retriever and report the performance in MRR@10.

Large-scale Retrieval Task. Large-scale retrieval uses a retriever to fetch top-relevant passages from a collection. Here, a ranker only plays a teaching role for knowledge distillation into a bi-encoder based retriever. The retriever is then used to perform this task on MS-Marco dataset, where MRR@10 and R@50 are used as metrics.

4.1 Main Results

BM25-Reranking on MS-Marco Dev. The results of BM25-reranking on MS-Marco Dev are listed in Table 1. Based on BM25-retrieved top-1000 candidates (w/ an evaluation metric of 85.7% Recall@1000), it is shown that the proposed R^{2}ANKER outperforms the best previously reported result, 40.1% MRR@10, from RocketQAv2, and delivers state-of-the-art BM25 reranking performance with 41.1 % MRR@10.

BM25-Reranking on TREC Deep Learning 2019. Furthermore, we also compare our method with a strong baseline (RocketQAv2) on TREC Deep Learning 2019 dataset for BM25-reranking. As shown in Table 2, it is observed that our method significantly outperforms RocketQAv2 by absolute 2.6% on NDCG@10. It further demonstrates the effectiveness of our approach.

Full Ranking. To comprehensively verify the effectiveness of R^{2}ANKER, we compare our method with other strong baselines with different retrievers other than BM25. As shown in the second part of Table 1, our method outperforms other models by about 1.4% $\sim$ 2.4% on MRR@10. Moreover, we can observe that, our R^{2}ANKER still performs better than the ranker in RocketQA and RocketQAv2 even though ours is associated with the weakest retriever (i.e., coCondenser† whose R@50=83.5%), demonstrating superiority of our R^{2}ANKER.

Large-scale Retrieval. Recently, using a ranker for knowledge distillation into a retriever becomes a prevalent technique for large-scale retriever (Ren et al., 2021b; Zhang et al., 2022a). This distillation process can also be employed to evaluate a ranker based on whether the ranker can provide valid and effective relevance scores for retriever training. As shown in Table 3, we integrated our R^{2}ANKER into the training pipeline of two popular dense-vector retrievers, i.e., coCondenser and SimLM. It is observed that compared to the methods (i.e., coCondenser and AR2) with coCondenser as initialization, the retriever distilled from our R^{2}ANKER significantly upgrades: i) compared to coCondenser w/o distillation, our distillation improves MRR@10 by 1.6%; and ii) compared to AR2 co-training even with a retriever-specific ranker, our distilled retriever can offer a 0.5% MRR@10 lift. On the other hand, we also integrated our ranker into a state-of-the-art bottleneck pre-training framework, ED-MLM (Wang et al., 2022), which demonstrates our method is capable of surpassing the previous carefully-designed retrieval methods. Moreover, it is noteworthy that

Ranker	Retriever	#Cand&Recall	MRR@10
BM25-Reranking			
BM25 (Yang et al., 2017)	BM25	1000 (85.7)	18.7
$\text{BERT}_{\text{base}}$ (Qiao et al., 2019)	BM25	1000 (85.7)	33.7
ColBERT (Khattab and Zaharia, 2020)	BM25	1000 (85.7)	34.9
SAN + $\text{BERT}_{\text{base}}$ (Liu et al., 2018)	BM25	1000 (85.7)	37.0
RocketQA (Qu et al., 2021)	BM25	1000 (85.7)	37.0
Multi-stage (Nogueira et al., 2019)	BM25	1000 (85.7)	39.0
CAKD (Hofstätter et al., 2020)	BM25	1000 (85.7)	39.0
RocketQAv2 (Ren et al., 2021b)	BM25	1000 (85.7)	40.1
R^2ANKER (Ours)	BM25	1000 (85.7)	**41.1**
Full-Ranking			
RocketQA (Qu et al., 2021)	RocketQA (Qu et al., 2021)	50 (85.5)	40.9
RocketQAv2 (Ren et al., 2021b)	RocketQA (Qu et al., 2021)	50 (85.5)	41.8
RocketQAv2 (Ren et al., 2021b)	RocketQAv2 (Ren et al., 2021b)	50 (86.2)	41.9
R^2ANKER (Ours)	coCondenser† (Gao and Callan, 2022)	50 (83.5)	42.7
R^2ANKER (Ours)	coCondenser§ (Gao and Callan, 2022)	50 (86.4)	**43.3**
R^2ANKER (Ours)	SPLADE† (Formal et al., 2021)	50 (84.3)	43.0
R^2ANKER (Ours)	SPLADE§ (Formal et al., 2021)	50 (86.1)	**43.3**

Table 1: BM25-reranking and full-ranking results on MS-Marco Dev. '#Cand&Recall' denotes the number of retriever-provided top candidates for reranking, as well as the retriever's top-N recall metric (%). † denotes the retriever trained on BM25 negatives, whereas § denotes the retriever trained on hard negatives sampled by its corresponding † retriever.

Method	NDCG@10
RocketQAv2 (Ren et al., 2021b)	71.4
R^2ANKER (Ours)	**73.0**

Table 2: BM25-reranking results on TREC 2019.

our R^2ANKER is not retriever specific and does not depend on retriever-ranker co-training, making it flexible and applicable enough to any retriever training pipeline for superior performance.

4.2 Various Negative Generators

First of all, we need to detail more about the retrievers, i.e., coCondenser (Den) and SPLADE (Lex), involved in this work. In particular, training both retrievers following the pipeline Gao and Callan (2022), where the model is first trained over BM25 negatives (i.e., Den-BN & Lex-BN) and then continually trained over self-adversarial hard negatives (i.e., Den-HN & Lex-HN). In contrast to the mere use of Den-HN & Lex-HN in the above main results, we also involve the first-stage retrievers for extensive analyses. In the remainder, D1, D2, L1, and L2 denote Den-BN, Den-HN, Lex-BN, and Lex-HN retrievers, respectively.

Ranker-Retrievers Combinations. In Table 4, we first split the results into two parts: given various retrievers (i.e., the columns), we compare the re-ranking performance with different model structures – the bi-encoder based retriever and cross-encoder based ranker. It is obvious that any trained ranker beats all the retrievers by a large margin, which explains why the hard negatives generated by one single retriever cannot effectively fool a ranker. Meanwhile, we find that in contrast to stacking more retrievers, the best strategy to sample negatives and train a ranker is combining the best from every world (i.e., BM25+D2+L2), which achieves the best re-ranking results upon various retrievers.

Adversary w/ Stronger Retrievers. To check whether using more sophisticated retrievers as generators could improve the multi-adversarial process and boost the ranker's performance, we introduce two latest retrievers (before ranker-distillation), i.e., ED-MLM (Wang et al., 2022) for dense-vector retrieval (+1.3% cf. D2) and LexMAE (Shen et al., 2022) for lexicon-weighting retrieval (+2.0% cf. L2), for negative mining. But, as listed in Table 5, the two trained rankers achieve very competitive results, demonstrating that our method is not sensitive to retrievers' performance but their types.

4.3 Training-test Distribution Shift

A key evaluation metric for rankers is BM25 reranking, where BM25 retrieval is used to provide model-agnostic top-1000 negatives for reranking. Due to generality of BM25, it is more likely its top-1000 results include the hard negatives (even if few) that are also considered hard for an arbitrary retriever (Karpukhin et al., 2020; Chen et al., 2021). Therefore, we would like to figure out the correlation between divergence of training-test dis-

Method	Pre-train	Teacher	Specific?	Co-train?	MRR@10	R@50
BM25 (Yang et al., 2017)	-	-	-		18.7	59.2
ANCE (Xiong et al., 2021)	$\text{RoBERTa}_{\text{base}}$	-	-		33.0	-
ColBERT (Khattab and Zaharia, 2020)	$\text{BERT}_{\text{base}}$	-	-		36.0	82.9
RocketQA (Qu et al., 2021)	$\text{ERNIE}_{\text{base}}$	$\text{ERNIE}_{\text{base}}$	✓	✓	37.0	85.5
COIL (Gao et al., 2021)	$\text{BERT}_{\text{base}}$	-	-		35.5	-
ME-BERT (Luan et al., 2021)	$\text{BERT}_{\text{large}}$	-	-		33.8	-
PAIR (Ren et al., 2021a)	$\text{ERNIE}_{\text{base}}$	-	-		37.9	86.4
DPR-PAQ (Oguz et al., 2021)	$\text{BERT}_{\text{base}}$	-	-		31.4	-
Condenser (Gao and Callan, 2021)	$\text{Condenser}_{\text{base}}$	-	-		36.6	-
coCondenser (Gao and Callan, 2022)	$\text{coCondenser}_{\text{base}}$	-	-		38.2	-
RocketQAv2 (Ren et al., 2021b)	$\text{RocketQA}_{\text{base}}$	$\text{ERNIE}_{\text{base}}$	✓	✓	38.8	86.2
AR2 (Zhang et al., 2022a)	$\text{coCondenser}_{\text{base}}$	$\text{ERNIE}_{\text{large}}$	✓	✓	39.5	87.8
ERNIE-Search (Lu et al., 2022)	$\text{ERNIE}_{\text{base}}$	$\text{ERNIE}_{\text{large}}$	✓	✓	40.1	87.7
SimLM (Wang et al., 2022)	$\text{SimLM}_{\text{base}}$	$\text{ELECTRA}_{\text{base}}$	✓		41.1	87.8
Ours	$\text{coCondenser}_{\text{base}}$	$\text{R}^2\text{ANKER}_{\text{base}}$			40.0	87.6
Ours	$\text{ED-MLM}_{\text{base}}$	$\text{R}^2\text{ANKER}_{\text{base}}$			**41.4**	**88.6**

Table 3: First stage retrieval performance on MS-Marco Dev, where a non-empty 'Teacher' column denotes that the retriever is trained with a ranker. The 'Co-Train' denotes the ranker is also updated during training, otherwise frozen. The 'Specific' denotes whether the ranker is (co-)trained for a specific retriever, and note that all the rankers in our competitors are also fully trained.

Retriever	BM25	D1	D2	L1	L2
Retriever as reranker.					
BM25	21.23	22.50	21.90	21.61	21.58
Den-BN (**abbr.** D1)	35.51	36.15	36.14	36.28	36.24
Den-HN (**abbr.** D2)	36.76	38.12	38.12	38.14	38.14
Lex-BN (**abbr.** L1)	35.31	36.17	36.20	36.11	36.13
Lex-HN (**abbr.** L2)	36.86	38.24	38.26	38.18	38.18
Our ranker trained with retriever(s) as reranker.					
R^2ANKER					
- BM25	39.82	41.38	41.44	41.39	41.41
- D1	40.50	42.81	42.82	42.78	42.82
- D2	40.47	42.62	42.67	42.60	42.62
- L1	39.81	41.56	41.56	41.92	41.77
- L2	40.78	41.51	41.60	42.92	42.88
- D1,D2	40.71	42.96	43.00	42.93	42.94
- L1,L2	40.44	42.19	42.03	42.93	42.81
- D1,L1	40.70	42.98	42.92	42.98	42.98
- D2,L2	40.82	42.88	42.92	42.89	42.92
- BM25,D2,L2	**41.12**	**43.24**	**43.26**	**43.28**	**43.29**
- D1,L1,D2,L2	41.00	43.21	43.22	43.21	43.24
- BM25,D1,L1,D2,L2	40.78	42.99	42.95	42.97	42.99

Table 4: Full-ranking results in terms of MRR@10 by different retriever-ranker combinations.

Negative	MRR@10
BM25,D2,L2	41.1
BM25,D',L'	41.0

Table 5: BM25 reranking results with a stronger negative generator. D' & L' denotes ED-MLM & LexMAE, respectively.

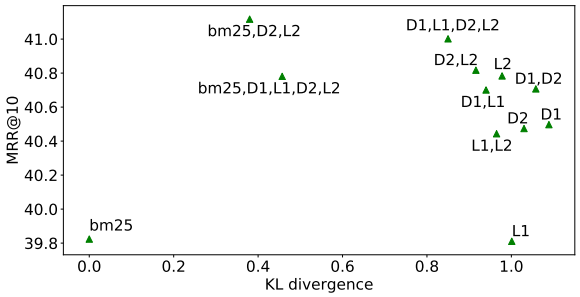


Figure 2: BM25-reranking performance by various rankers that were trained on negatives sampled from retrievers' (joint) distributions. 'KL divergence' denotes the difference between the retrievers' (joint) distribution and BM25 retriever's, i.e., $KL(P(\cdot|q;\Theta^{(\text{be})})|\,\text{BM25}(\cdot|q;\mathcal{D}))$, which is used to measure negatives' distribution. For example, the point 'bm25,D2,L2' denotes that i) the KL between its joint retriever's distribution and BM25 retriever's distribution is round 0.4, and ii) a ranker trained on that joint negative distribution can achieve 41.1 MRR@10 on BM25 reranking.

tributions and performance of final trained rankers, where the training distribution is generated by an adversarial generator upon one or more retrievers.

Correlation of BM25 test w/ Adversarial Train. Straightforwardly, the divergence can be easily calculated by applying a discrete KL divergence between the top retrieved results from a (joint) retriever and the BM25. As shown in Figure 2, the ranker achieves roughly better performance when the distribution of its training data is closer to that of test data, except for the BM25. This is possibly because i) the consistency of training-test data distribution avoids the distribution shift problem and achieves greater performance, and ii) though training a ranker on BM25 negatives seems perfect in distribution matching, the negatives are not challenging enough for effective training (Zhan et al., 2021; Ren et al., 2021b; Zhang et al., 2022a).

BM25-Constrained Negative Mining. To avoid the trivial (non-challenging) negatives from only BM25 and further investigate the impact of BM25-

Negatives to Train Ranker	Train MRR@10	Dev MRR@10	Δ
BM25	46.9	39.8	-7.1
BM25,D2,L2	**48.7**	**41.1**	-7.6
BM25 ∩ (D2,L2)	**48.7**	40.6	-8.1

Table 6: BM25-dependent negative sampling for ranker training, where the last denotes BM25-constrained sampling.

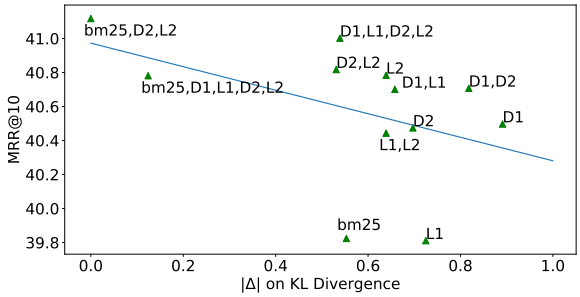


Figure 3: BM25-reranking performance of rankers *vs.* relevance score distribution changes from the negative generator to the trained ranker. As such, the smaller $|\Delta|$, the negative sampling distribution closer to the ideal negative distribution of the ranker, as learning on the retriever-sampled negatives will not shift the distribution. In formal, upon Figure 2, $\Delta = KL(P(\cdot|q;\Theta^{(\text{be})})|\,\text{BM25}(\cdot|q;\mathcal{D})) - KL(P(\cdot|q;\theta^{(\text{ce})})|\,\text{BM25}(\cdot|q;\mathcal{D}))$, where $\theta^{(\text{ce})}$ is trained with a specific $\Theta^{(\text{be})}$.

sourced negatives for ranker training, we present a BM25-constrained negative mining strategy. Here, a negative document, sampled from a generator comprised of D2 and L2, will be kept only if it also appears in BM25 top-1000 results. As such, the resulting training samples for the ranker would not be as trivial as those based on top BM25 negatives, where the trained ranker is prone to distinguish hard negatives in BM25 reranking. However, the training and dev results shown in Table 6 demonstrate that i) consistent with the above paragraph, the ranker trained on BM25 is likely to be under-fitting and ii) compared to a joint of diverse retrievers as the negative generator (i.e., BM25,D2,L2), focusing only on test-related BM25-constrained negatives leads to more severe over-fitting problem. This is likely because BM25 top-1000 reranking is general to evaluate a ranker trained on any negative distribution, and its key is to involve diverse, challenging negatives to make the ranker robust.

But, open questions remain about what kinds of negative sampling distribution matter and whether sampling in-distribution of a ranker leads to better performance, which we will answer in the next.

Negative	MRR@10
BM25,D2,L2	41.1
BM25,D1,D2,L1,L2	40.8
Re-dist(BM25 ∪ D1 ∪ D2 ∪ L1 ∪ L2; θ^{ce})	39.6

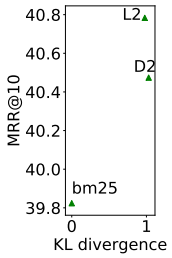
Table 7: Comparison of BM25 reranking with a ranker trained with re-distributed hard negatives (i.e., the 3rd one).

4.4 Ranker-aware Sampling Distribution

As in Introduction, the in-distribution negative sampling strategy has been proven effective in retrieval model training but is non-applicable to ranker training because of the combinatorial explosion. Therefore, we propose to approximately analyze the impact of in-distribution sampling for the ranker.

Correlation of Ranker w/ Negative Generator. As shown in Figure 3, we propose an approximate calculation method, which leverages the general BM25 results as mediums, to compare the distributions between a negative generator and a trained ranker. It is observed that there is a clear negative correlation between distance of the two distributions and performance of the corresponding trained ranker. This demonstrates that a generator could achieve more robust performance roughly when its distribution is more similar to that of a ranker.

Ranker-redistributed Negative Sampling. Although sampling negatives from the in-distribution of a ranker is practically impossible due to the combinatorial explosion, we present a re-distribution strategy to simulate the in-distribution. Specifically, we first leverage all the retrievers to retrieve top-1000 negatives from the collection individually and then combine them into a negative pool for a query. Next, we apply a BM25-trained ranker to the pool for self-adversarial sampling. Last, we employ the sampled negatives to train a new ranker and report its result in Table 7. As we can see, the re-distribution result is the worst among the three models despite its promising in terms of the negatives' difficulty. This is because the top candidates by the strong ranker are full of false negatives (this explains why some previous works use a ranker to de-noise (Qu et al., 2021; Formal et al., 2021)), verifying the ineffectiveness of in-distribution negatives for ranker training. In contrast, our multi-adversarial training strategy by a joint generator can provide mutually out-of-distribution negatives and thus benefits from open-domain noises for robust training (Wei et al., 2021).



Query	*ID:72809//* can prozac cause nausea	*ID:189466//* foods that help cortisol stress	*ID:284072//* how many gallons is liters
False Negative	*ID:5467837//* Side effects associated with Prozac include headache, nervousness, agitation, insomnia, tremor, fatigue, rash, lightheadedness, drowsiness, dizziness, depersonalization, nausea, upset stomach, diarrhea, loss of appetite, stomach pain, sweating and dry mouth.	*ID:6115376//* How to Lower High Cortisol Levels Naturally<sep>Ways to Lower Cortisol Step 1. Eat a balanced, nutritious diet that contains plenty of fruits and vegetables. Low-glycemic index foods like eggs help lower cortisol levels in the blood while whole grain products help proteins control the production of cortisol in the body. Stay away from processed sugars and flours. Vitamin B5 and folic acid help lower cortisol levels.	*ID:6134923//* What Is the Equivalent of 1 Liter in Gallons?<sep>Quick Answer. One liter is the equivalent of approximately 0.2642 gallons. As both terms are measurements of volume, one can convert between the two by utilizing the fact that there are about 3.785 liters per gallon.
Rank	*D2*: 5; *L2*: 7; *BM25*: 65	*D2*: 8; *L2*: 33; *BM25*: 131	*D2*: 20; *L2*: 1; *BM25*: 207

Figure 4: BM25-reranking performance of the rankers trained on different negative distributions by specific retrievers (left) and false negative labels brought by the two well-trained strong retrievers in contrast to BM25 retriever (right).

4.5 Case Study

To investigate how to learn an effective ranker, we show the results of rankers trained from three retrievers, i.e., BM25, D2, and L2. In contrast to well-trained D2 and L2, we can see the BM25 negatives cannot challenge a high-capable ranker built upon pre-trained language models with cross-encoder structure, leading to sub-optimal results (see the BM25 point in Figure 4(left)). However, a strong well-trained retriever as the negative sampler is more likely to introduce label noises, a.k.a. false negative labels in the retrieval field (Qu et al., 2021). Please refer to Figure 4(right) for several examples. Therefore, we leverage multiple retrievers as the generator to learn an effective ranker, where extensive out-of-distribution label noises from retrievers render the ranker against each noise distribution.

5 Conclusion

In this work, we propose a multi-adversarial strategy for robust ranker training where a negative generator built upon a joint of diverse retrievers is proposed to sample challenging hard negatives for effective adversarial learning. Empirically, our proposed ranker, R^{2}ANKER, achieves state-of-the-art performance in both BM25-reranking and full-ranking tasks on benchmark datasets. And empowered by our R^{2}ANKER as a teacher for distillation, previous basic retrievers can now deliver state-of-the-art results in large-scale retrieval tasks. Moreover, our insightful analysis also reveals the minor impact of training-test distribution shift in BM25 reranking due to the generality of BM25 retriever. Meantime, we also find that there is a negative correlation between the ranker-generator distance and performance of the ranker, and re-distribution towards a ranker is toxic due to false negative labels.

Limitations

The limitations of our R^{2}ANKER includes i) *Performance Bottleneck*: As verified in our experiments, the performance of multi-adversarial ranker training depends more on types of the comprising retrievers than their performance. Since the number of the types is very limited, there is a performance bottleneck of our method. and ii) *Compromised Adversary*: Due to computation overheads, the adversarial process is compromised in our training framework in terms of real-time retriever updating. This would negatively affect the performance of the framework.

References

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 233–242. PMLR.

Dan Brickley, Matthew Burgess, and Natasha F. Noy. 2019. Google dataset search: Building a search engine for datasets in an open web ecosystem. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 1365–1375. ACM.

Xilun Chen, Kushal Lakhotia, Barlas Oguz, Anchit Gupta, Patrick S. H. Lewis, Stan Peshterliev, Yashar Mehdad, Sonal Gupta, and Wen-tau Yih. 2021. Salient phrase aware dense retrieval: Can a dense retriever imitate a sparse one? *CoRR*, abs/2110.06918.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. *CoRR*, abs/2003.07820.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE v2: Sparse lexical and expansion model for information retrieval. *CoRR*, abs/2109.10086.

Luyu Gao and Jamie Callan. 2021. Condenser: a pre-training architecture for dense retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 981–993. Association for Computational Linguistics.

Luyu Gao and Jamie Callan. 2022. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2843–2853. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. COIL: revisit exact lexical match in information retrieval with contextualized inverted list. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3030–3042. Association for Computational Linguistics.

Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. 2022. Semantic models for the first-stage retrieval: A comprehensive review. *ACM Trans. Inf. Syst.*, 40(4):66:1–66:42.

Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving efficient neural ranking models with cross-architecture knowledge distillation. *CoRR*, abs/2010.02666.

Yen-Chang Hsu, Zhaoyang Lv, Joel Schlosser, Phillip Odom, and Zsolt Kira. 2019. Multi-class classification without multi-class labels. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

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, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Xiaodong Liu, Kevin Duh, and Jianfeng Gao. 2018. Stochastic answer networks for natural language inference. *CoRR*, abs/1804.07888.

Yuxiang Lu, Yiding Liu, Jiaxiang Liu, Yunsheng Shi, Zhengjie Huang, Shikun Feng, Yu Sun, Hao Tian, Hua Wu, Shuaiqiang Wang, Dawei Yin, and Haifeng Wang. 2022. Ernie-search: Bridging cross-encoder with dual-encoder via self on-the-fly distillation for dense passage retrieval. *CoRR*, abs/2205.09153.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Trans. Assoc. Comput. Linguistics*, 9:329–345.

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 *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Rodrigo Frassetto Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *CoRR*, abs/1901.04085.

Rodrigo Frassetto Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *CoRR*, abs/1910.14424.

Barlas Oguz, Kushal Lakhotia, Anchit Gupta, Patrick S. H. Lewis, Vladimir Karpukhin, Aleksandra Piktus, Xilun Chen, Sebastian Riedel, Wen-tau Yih, Sonal Gupta, and Yashar Mehdad. 2021. Domain-matched pre-training tasks for dense retrieval. *CoRR*, abs/2107.13602.

Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the behaviors of BERT in ranking. *CoRR*, abs/1904.07531.

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, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5835–5847. Association for Computational Linguistics.

Ruiyang Ren, Shangwen Lv, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021a. PAIR: leveraging passage-centric similarity relation for improving dense passage retrieval. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2173–2183. Association for Computational Linguistics.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021b. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 2825–2835. Association for Computational Linguistics.

Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Xiaolong Huang, Binxing Jiao, Linjun Yang, and Daxin Jiang. 2022. Lexmae: Lexicon-bottlenecked pretraining for large-scale retrieval. *CoRR*, abs/2208.14754.

Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: enhanced representation through knowledge integration. *CoRR*, abs/1904.09223.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Simlm: Pre-training with representation bottleneck for dense passage retrieval. *CoRR*, abs/2207.02578.

Hongxin Wei, Lue Tao, Renchunzi Xie, and Bo An. 2021. Open-set label noise can improve robustness against inherent label noise. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 7978–7992.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 1253–1256. ACM.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 1503–1512. ACM.

Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2022a. Adversarial retriever-ranker for dense text retrieval.

Kai Zhang, Chongyang Tao, Tao Shen, Can Xu, Xiubo Geng, Binxing Jiao, and Daxin Jiang. 2022b. LED: lexicon-enlightened dense retriever for large-scale retrieval. *CoRR*, abs/2208.13661.

Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.*, 52(1):5:1–5:38.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Yucheng Zhou, Tao Shen, Xiubo Geng, Chongyang Tao, Guodong Long, Can Xu, and Daxin Jiang. 2022. Fine-grained distillation for long document retrieval. *CoRR*, abs/2212.10423.

A From the View of Open-set Noise

In this section, we introduce noise-labeling problem caused by false-negative samples, and then elaborate on why our proposed multi-adversarial learning strategy can mitigate the problem by open-set noise (Wei et al., 2021) and improve robustness.

Due to limited crowd-sourcing resources, it is impossible to comprehensively annotate the relevance of every query $\forall q \in \mathcal{Q}^{(\text{trn})}$ to every document $\forall d_+^q \in \mathbb{D}$. In general, the annotating process can be roughly described as i) using the best on-hand retriever (e.g., a commercial search engine) to fetch top document candidates for a query q, and then ii) distinguish positive document(s), d_+, associated to q from the very top candidates. Therefore, constrained by the retriever in the annotation process,

there exists positive documents for q not included in the top candidates, which are regarded as negative by mistake – false negative label – degrading standard ranker training. As such, sampling hard negatives by a strong retriever usually introduces the label-noise problem. Prior works focus on 'co-teaching' or/and 'boosting' strategies (Qu et al., 2021; Zhang et al., 2022a), but they assume a ranker is robust enough for anti-noise while only denoise for more fragile retrievers by the ranker.

Taking a step further, we could also formulate the search problem (both retrieval and rerank) as a many-class many-label classification problem, where the number of classes equals to $|\mathbb{D}|$, i.e., the number of documents in $\mathbb{D}$. And $|\mathbb{D}|$ is usually very large, ranging from millions to billions. Thus, the current solutions of the search problem are analogous to *label semantic matching* paradigm for many-class classification problems (Hsu et al., 2019). As such, the mis-labeled class caused by a single $\theta_j^{(\text{be})}$-parameterized retriever $\mathcal{R}_j$ will be subject to the following distribution:

$$y' \sim P^{(\text{FN})}(\text{d}|q, \mathbb{D}\backslash\{d_+\}; \theta_j^{(\text{be})}), \qquad (10)$$

where $P^{(\text{FN})}(\cdot|\cdot; \theta_j^{(\text{be})})$ denotes an inherent label noise distribution by the retriever $\theta_j^{(\text{be})}$.

Fortunately, from the view of noise-labeling problem in many-class many-label classification, we could employ the concept of 'open-set noise' to verify robustness of our ranker. As proven by a recent "*insufficient capacity*" (Arpit et al., 2017) assumption[2], learning a variety of out-of-distribution (OOD) or open-set noise can improve robustness against inherent label noises that are subject to one dataset or one distribution. Therefore, as multiple heterogeneous retrievers are involved in our multi-adversarial training framework to provide mutually-OOD hard negatives, the ranker trained on these samples can be robust to the noises from every single retriever, resulting in a superior ranker after the training.

B Framework Grounding Detials

Retriever Training. In line with (Clark et al., 2020) and (Ren et al., 2021b), we do not seek for updating the generators (i.e., the retrievers in our method) w.r.t performance of the discriminator (i.e., the ranker) with two considerations: On the one hand, we try to avoid heavy computation overheads to train the retrievers jointly and update the large-scale index synchronously. On the other hand, due to their intrinsic discrepancy in model structure, the generators hardly fool the discriminator, making the adversarial process less effective. As verified in (Zhang et al., 2022a), cooperative learning (training the retriever towards the reranker, regularized by a Kullback–Leibler divergence) is also necessary for competitive performance.

Sampling Negatives. The strategy to sample a negative distribution in Eq.(3) plays an important role in our method. Instead of directly sampling from the $\mathrm{softmax}$ distribution over $\mathbb{D}\backslash\{d_+\}$ that inclining to the very top candidates, we follow the previously common practice to cap top-N (says N=200 in our experiments) candidates and then conduct a uniform sampling to ensure its diversity. As for sampling over the joint negative distribution in Eq.(9), we combine the capped top-N candidates from multiple generators (retrievers) without de-duplication to better simulate the joint distribution.

C Training Setup

Ranker Training. We use the ERNIE-2.0-en-base model (Sun et al., 2019) as the initialization of our ranker. To provide more diverse hard negatives for ranker's robust training, we sample them from multiple retrievers: we use three kinds of retrievers, including BM25 (Yang et al., 2017) for term-based retrieval, coCondenser (Gao and Callan, 2022) for dense-vector retrieval models and SPLADE (Formal et al., 2021) for lexicon-weighting retrieval models. During ranker training, we sample 40 hard negatives for each query. The maximum training epoch, batch size and learning rate are set to 2, 12 and 1×10^{-5}. The maximum sequence length is set to 128 and the random seed is fixed to 42. For model optimizing, we use Adam optimizer (Kingma and Ba, 2015) and a linear warmup. The warmup proportion is 0.1, and the weight decay is 0.1. All experiments are conducted on an A100 GPU.

Retriever Distillation. To distill our trained ranker to a retriever for first-stage retrieval, we adopt the two-stage coCondenser retriever (Gao and Callan, 2022) and apply our ranker scores

[2]By Wei et al. (2021), "increasing the number of examples while keeping representation capacity fixed would increase the time needed to memorize the data set. Hence, the larger the size of auxiliary dataset is, the more time it needs to memorize the open-set noises in the auxiliary dataset as well as the inherent noises in the training set, relative to clean data."

to the second stage of coCondenser fine-tuning. Specifically, instead of the mere contrastive learning, we leverage training data by Ren et al. (2021b) and replace contrastive learning loss in coCondenser with a simple KL divergence loss. Its learning rate, batch size, and epoch number are set to 5×10^{-5}, $16\times$ (1 positive and 10 negatives), and 4, respectively.

Retriever Pre-training. To make the distilled results more competitive, we also involve the recently sophisticated bottleneck pre-training technique, called ED-MLM (Wang et al., 2022). Upon an initialization from BERT-base (Devlin et al., 2019) and data from MS-Marco collection, the learning rate is set to 1×10^{-4}, the batch size is set to 2048, the number of training epochs is set to 20, max sequence length is set to 144, and the random seed is set to 42. The other parameters are strictly following Wang et al. (2022). Such a corpus-aware pre-training procedure takes about 13 hours on eight A100 GPUs.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations section

☒ A2. Did you discuss any potential risks of your work?
The topic of the paper deals only with text retrieval

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Introduction section

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☑ B1. Did you cite the creators of artifacts you used?
4 Experiments section

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
MS-Marco and TREC Deep Learning 2019 are open-source datasets

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Our use of MS-Marco and TREC Deep Learning 2019 was consistent with their intended use.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
4 Experiments section

C ☑ Did you run computational experiments?

4 Experiments section

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix D Training Setup

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix D Training Setup

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4 Experiments section

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4 Experiments section

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.