RocketQAv2: A Joint Training Method for Dense Passage Retrieval and Passage Re-ranking

Ruiyang Ren[1,3]*, Yingqi Qu[2]*, Jing Liu[2], Wayne Xin Zhao[3,4]†, Qiaoqiao She[2]
Hua Wu[2]†, Haifeng Wang[2] and Ji-Rong Wen[1,3,4]
[1]School of Information, Renmin University of China; [2]Baidu Inc.
[3]Beijing Key Laboratory of Big Data Management and Analysis Methods
[4]Gaoling School of Artificial Intelligence, Renmin University of China
{reyon.ren, jrwen}@ruc.edu.cn, batmanfly@gmail.com
{quyingqi, liujing46, sheqiaoqiao, wu_hua, wanghaifeng}@baidu.com

Abstract

In various natural language processing tasks, passage retrieval and passage re-ranking are two key procedures in finding and ranking relevant information. Since both the two procedures contribute to the final performance, it is important to jointly optimize them in order to achieve mutual improvement. In this paper, we propose a novel joint training approach for dense passage retrieval and passage re-ranking. A major contribution is that we introduce the dynamic listwise distillation, where we design a unified listwise training approach for both the retriever and the re-ranker. During the dynamic distillation, the retriever and the re-ranker can be adaptively improved according to each other's relevance information. We also propose a hybrid data augmentation strategy to construct diverse training instances for listwise training approach. Extensive experiments show the effectiveness of our approach on both MSMARCO and Natural Questions datasets. Our code is available at https://github.com/PaddlePaddle/RocketQA.

1 Introduction

Recently, dense passage retrieval has become an important approach in the task of *passage retrieval* (Karpukhin et al., 2020) to identify relevant contents from a large corpus. The underlying idea is to represent both queries and passages as low-dimensional vectors (a.k.a., embeddings), so that the relevance can be measured via embedding similarity. Additionally, a subsequent procedure of *passage re-ranking* is widely adopted to further improve the retrieval results by incorporating a re-ranker (Qu et al., 2021; Luan et al., 2021). Such a two-stage procedure is particularly useful in a variety of natural language processing tasks, including question answering (Mao et al., 2021; Xiong et al., 2020b), dialogue system (Ji et al., 2014; Henderson et al., 2017) and entity linking (Gillick et al., 2019; Wu et al., 2020).

Following a *retrieve-then-rerank* way, the dense retriever in passage retrieval and the re-ranker in passage re-ranking jointly contribute to the final performance. Despite the fact that the two modules work as a pipeline during the inference stage, it has been found useful to train them in a correlated manner. For example, the retriever with a dual-encoder can be improved by distilling from the re-ranker with a more capable cross-encoder architecture (Qu et al., 2021; Yang et al., 2020), and the re-ranker can be improved with training instances generated from the retriever (Qu et al., 2021; Huang et al., 2020; Gao et al., 2021b). Therefore, there is increasing attention on correlating the training of the retriever and re-ranker in order to achieve mutual improvement (Metzler et al., 2021; Qu et al., 2021; Huang et al., 2020; Yang et al., 2020). Typically, these attempts train the two modules in an alternative way: fixing one module and then optimizing another module. It will be more ideal to mutually improve the two modules in a joint training approach.

However, the two modules are usually optimized in different ways, so that the joint learning cannot be trivially implemented. Specially, the retriever is usually trained by sampling a number of in-batch negatives to maximize the probabilities of positive passages and minimize the probabilities of the sampled negatives (Xiong et al., 2020a; Karpukhin et al., 2020), where the model is learned by considering the entire list of positive and negatives (called *listwise* approach[1]). As a comparison, the re-ranker is usually learned in a pointwise or pairwise manner (Nogueira and Cho, 2019; Nogueira et al., 2019b), where the model is

* Equal contribution. The work was done when Ruiyang Ren was doing internship at Baidu.

† Corresponding authors.

[1] Instead of considering the total order as in learning to rank (Cao et al., 2007), we use "listwise" to indicate that relevance scores are derived based on a candidate list.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2825–2835
November 7–11, 2021. ©2021 Association for Computational Linguistics

learned based on a single passage or a pair of passages. To address this issue, our idea is to unify the learning approach for both retriever and re-ranker. Specially, we adopt the listwise training approach for both retriever and re-ranker, where the relevance scores are computed according to a list of positive and negative passages. Besides, it is expected to include diverse and high-quality training instances for the listwise training approach, which can better represent the distribution of all the passages in the whole collection. Thus, it requires more effective data augmentation to construct the training instances for listwise training.

To this end, we present a novel joint training approach for dense passage retrieval and passage re-ranking (called **RocketQAv2**). The major contribution of our approach is the novel *dynamic listwise distillation* mechanism for jointly training the retriever and the re-ranker. Based on a unified listwise training approach, we can readily transfer relevance information between the two modules. Unlike previous distillation methods that usually froze one module, our approach enables the two modules to adaptively learn relevance information from each other, which is the key to mutual improvement in joint training. Furthermore, we design a hybrid data augmentation strategy to generate diverse training instances for listwise training approach.

The contributions of this paper can be summarized as follows:

- We propose a novel approach that jointly trains the dense passage retriever and passage re-ranker. It is the first time that joint training has been implemented for the two modules.

- We make two major technical contributions by introducing dynamic listwise distillation and hybrid data augmentation to support the proposed joint learning approach.

- Extensive experiments show the effectiveness of our proposed approach on both MSMARCO and Natural Questions datasets.

2 Related Work

Recently, dense passage retrieval has demonstrated better performance than traditional sparse retrieval methods (e.g., TF-IDF and BM25) on the task of passage retrieval. Existing approaches of learning dense passage retriever can be divided into two categories: (1) self-supervised pretraining for retrieval (Chang et al., 2020; Lee et al., 2019; Guu et al., 2020) and (2) fine-tuning pre-trained language models (PLMs) on labeled data (Lu et al., 2020; Karpukhin et al., 2020; Xiong et al., 2020a; Luan et al., 2021; Qu et al., 2021) . Our work follows the second class of approaches, which show better performance with less cost. There are two important tricks to train an effective dense retriever: (1) incorporating hard negatives during training (Karpukhin et al., 2020; Xiong et al., 2020a; Qu et al., 2021) and (2) distilling the knowledge from cross-encoder-based reranker into dual-encoder-based retriever (Izacard and Grave, 2020; Yang and Seo, 2020; Qu et al., 2021; Ren et al., 2021). Based on the retrieved passages from a retriever, PLM-based rerankers with the cross-encoder architecture have recently been applied on passage re-ranking to improve the retrieval results (Qiao et al., 2019; Nogueira and Cho, 2019; Wang et al., 2019; Yan et al., 2019), and yield substantial improvements over the traditional methods.

Apart from separately considering the above two tasks, it has been proved that passage retrieval and passage re-ranking are actually highly related and dependent (Huang et al., 2020; Gao et al., 2020; Khattab and Zaharia, 2020). The retriever needs to capture the relevance knowledge from the re-ranker, and the re-ranker should be specially optimized according to the preceding results of the retriever. Some efforts studied the possibility of leveraging the dependency of retriever and re-ranker, and try to enhance the connection between them in an alternative way (Qu et al., 2021; Yang et al., 2020; Huang et al., 2020). Furthermore, several studies attempted to jointly train the retriever and the reader for Open-domain Question Answering (Guu et al., 2020; Sachan et al., 2021; Karpukhin et al., 2020). Different from the prior studies, our method is a joint learning architecture of the dense passage retriever and the re-ranker.

3 Methodology

In this section, we describe a novel joint training approach for dense passage retrieval and passage re-ranking (called **RocketQAv2**)

3.1 Overview

In this work, we consider two tasks including dense passage retrieval and passage re-ranking, which are described as follows.

Given a query q, the aim of *dense passage retrieval* is to retrieve k most relevant passages from a large collection of M text passages. The dual-encoder (DE) architecture is widely adopted by prior works (Karpukhin et al., 2020; Luan et al., 2021; Qu et al., 2021), where two separate dense encoders $E_P(\cdot)$ and $E_Q(\cdot)$ are used to map passages and queries to d-dimensional real-valued vectors (a.k.a., embeddings) separately, and then an index of all passage embeddings is built for efficient retrieval. The similarity between the query q and the passage p is defined using the dot product:

$$s_{\mathrm{de}}(q,p) = E_Q(q)^\top \cdot E_P(p). \qquad (1)$$

Given a list of candidate passages retrieved by a passage retriever, the aim of *passage re-ranking* is to further improve the retrieval results with a re-ranker, which estimates a relevance score $s(q,p)$ measuring the relevance level of a candidate passage p to a query q. Among the implementations of the re-ranker, a cross-encoder (CE) based on PLMs usually achieves superior performance (Nogueira and Cho, 2019; Qiao et al., 2019), which can better capture the semantic interactions between the passage and the query, but requires more computational efforts than the dual-encoder. To compute the relevance score $s_{\mathrm{ce}}(q,p)$, a special token [SEP] is inserted between q and p, and then the representation at the [CLS] token from the cross-encoder is fed into a learned linear function.

Usually, the passage retriever and the passage re-ranker are learned in either a separate or alternative way (*i.e.,* fixing one and then training the other). To achieve the joint training, we introduce *dynamic listwise distillation* (Section 3.2), which can adaptively improve both components in a joint optimization process. To support the listwise training, we further propose *hybrid data augmentation* (Section 3.2) for generating diverse and high-quality training instances. Based on the two major contributions, we present the learning procedure in Section 3.4 and related discussion in Section 3.5.

3.2 Dynamic Listwise Distillation

Since the re-ranker adopts the more capable cross-encoder architecture, it has become a common strategy to distill the knowledge from re-ranker into the retriever. However, in prior studies (Karpukhin et al., 2020; Xiong et al., 2020a; Qu et al., 2021), the retriever and re-ranker are usually learned in different ways, and the parameters of the re-ranker are *frozen*, which cannot jointly optimize the two components for mutual improvement. Considering this issue, we design a unified listwise training approach to learn both the retriever and the re-ranker, and dynamically update both the parameters of the re-ranker and the retriever during distillation. In this way, the two components can adaptively improve each other. We call this approach as *dynamic listwise distillation*. Next, we will describe the details of *dynamic listwise distillation*.

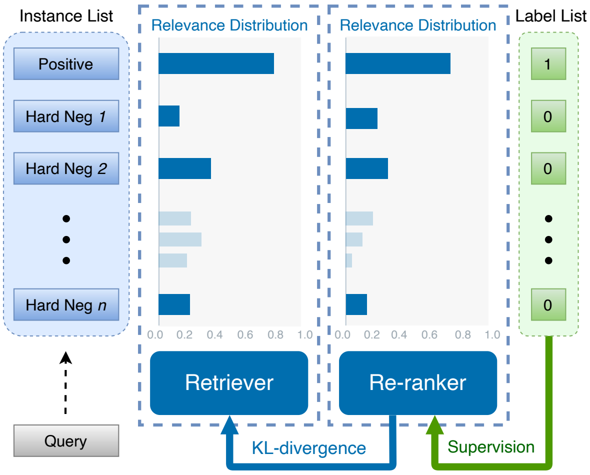

Figure 1: The illustration of dynamic listwise distillation in our approach.

Formally, given a query q in a query set $\mathcal{Q}$ and the corresponding list of candidate passages (instance list) $\mathcal{P}_q = \{p_{q,i}\}_{1\leq i\leq m}$ related to query q, we can obtain the relevance scores $S_{\mathrm{de}}(q) = \{s_{\mathrm{de}}(q,p)\}_{p\in\mathcal{P}_q}$ and $S_{\mathrm{ce}}(q) = \{s_{\mathrm{ce}}(q,p)\}_{p\in\mathcal{P}_q}$ of a query q and passages in $\mathcal{P}_q$ from the dual-encoder-based retriever and the cross-encoder-based re-ranker, respectively. Then, we normalize them in a listwise way to obtain the corresponding relevance distributions over candidate passages:

$$\tilde{s}_{\mathrm{de}}(q,p) = \frac{e^{s_{\mathrm{de}}(q,p)}}{\sum_{p'\in\mathcal{P}_q} e^{s_{\mathrm{de}}(q,p')}}, \qquad (2)$$

$$\tilde{s}_{\mathrm{ce}}(q,p) = \frac{e^{s_{\mathrm{ce}}(q,p)}}{\sum_{p'\in\mathcal{P}_q} e^{s_{\mathrm{ce}}(q,p')}}. \qquad (3)$$

The main idea is to adaptively reduce the difference between the two distributions from the retriever and the re-ranker so as to mutually improve each other. To achieve the adaptively mutual improvement, we minimize the KL-divergence be-

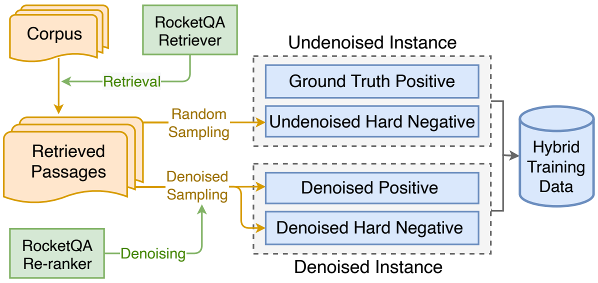

Figure 2: The illustration of hybrid data augmentation.

tween the two relevance distributions $\{\tilde{s}_{\text{de}}(q,p)\}$ and $\{\tilde{s}_{\text{ce}}(q,p)\}$ from the two modules:

$$\mathcal{L}_{\text{KL}} = \sum_{q\in\mathcal{Q},p\in\mathcal{P}_q} \tilde{s}_{\text{de}}(q,p)\cdot\log\frac{\tilde{s}_{\text{de}}(q,p)}{\tilde{s}_{\text{ce}}(q,p)}. \quad (4)$$

Additionally, we provide ground-truth guidance for the joint training. Specifically, we also adopt a cross-entropy loss for the re-ranker based on passages in $\mathcal{P}_q$ with supervised information:

$$\mathcal{L}_{\text{sup}} = -\frac{1}{N}\sum_{q\in\mathcal{Q},p^+}\log\frac{e^{s_{\text{ce}}(q,p^+)}}{e^{s_{\text{ce}}(q,p^+)}+\sum_{p^-}e^{s_{\text{ce}}(q,p^-)}}, \quad (5)$$

where N is the number of training instances, and p^+ and p^- denote the positive passage and negative passage in $\mathcal{P}_q$, respectively. We combine the KL-divergence loss and the supervised cross-entropy loss defined in Eq. (4) and Eq. (5) to obtain the final loss function:

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{sup}}. \quad (6)$$

Figure 1 presents the illustration of the dynamic listwise distillation. The re-ranker is optimized with labeled lists (Eq. (5)), and it produces relevance distributions to train the retriever (Eq. (4)). Unlike RocketQA that conducts hard pseudo labeled data (Qu et al., 2021), we utilize soft labels (*i.e.,* estimated relevance distributions) for relevance distillation. Besides, we dynamically update the parameters of the re-ranker in order to adaptively synchronize the two modules for mutual improvement. To discriminate from the previous static distillation based on pseudo labels, we call our method as *dynamic listwise distillation*.

3.3 Hybrid Data Augmentation

To perform dynamic listwise distillation, we need to generate the candidate passage list $\mathcal{P}_q$ for query q. Since our approach relies on listwise training, we expect the candidate passage list includes diverse and high-quality candidate passages, which may better represent the distribution of all the passages in the whole collection. Prior works (Xiong et al., 2020a; Qu et al., 2021; Karpukhin et al., 2020) demonstrate that it is important to include hard negatives in the candidate passage list. Basically, ANCE (Xiong et al., 2020a) and DRP (Karpukhin et al., 2020) introduces the randomly sampled hard negatives, while RocketQA (Qu et al., 2021) incorporates denoised hard negatives. Inspired by prior works, we design a hybird data augmentation way to construct diverse training instances by incorporating both random sampling and denoised sampling.

As shown in Figure 2, our proposed hybrid data augmentation includes both undenoised and denoised instances. First, we utilize the RocketQA retriever to retrieve top-n passages from the corpus. For undenoised instances, we randomly sample the undenoised hard negatives from retrieved passages and include ground-truth positives. For denoised instances, we utilize the RocketQA re-ranker to remove the predicted negatives with low confidence scores. We also include denoised positives that are predicted as positives by the RocketQA re-ranker with high confidence scores.

Compared with previous methods, our data augmentation method utilizes more ways (undenoised or denoised) to generate both positives and negatives to improve the diversity of instances list $\mathcal{P}_q$. Specially, we mainly focus on including hard negatives. This is particularly important to dynamic listwise distillation, since weak negatives are easy to be identified, which cannot increase additional gain for both modules.

3.4 Training Procedure

In this section, we present the training procedure of our approach.

Figure 3 presents the illustration of the training procedure for our approach. We first initialize the retriever and re-ranker with the learned dual-encoder and cross-encoder of RocketQA [2]. Then, we utilize the retriever and re-ranker in RocketQA

[2] Note that in this paper, RocketQA retriever is the model in the first step of RocketQA and the RocketQA re-ranker is the model in the second step of RocketQA. The two models can also be replaced with other trained retriever and re-ranker. We found that using the trained model to initialize retriever and reranker can help achieve slightly better results. This is due to the fact that the retriever and re-ranker have a mutual influence during training, the initialized retriever and re-ranker can facilitate the initial optimization stage.

Dataset	#query in train	#query in dev	#query in test	#passage
MSMARCO	502,939	6,980	6.837	8,841,823
Natural Questions	58,812	6,515	3,610	21,015,324

Table 1: The detailed statistics of MSMARCO and Natural Questions.

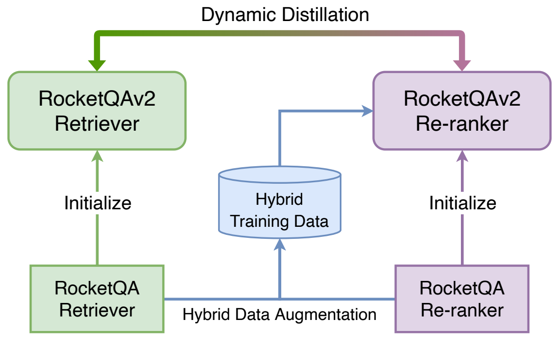

Figure 3: The overall joint training architecture of RocketQAv2.

to generate the training data via hybrid data augmentation in Section 3.3. Finally, we preform dynamic listwise distillation to jointly optimize the retriever and re-ranker following Section 3.2. During distillation, the retriever and re-ranker are mutually optimized according to the final retrieval performance. After the training stage, we can apply the retriever and re-ranker for inference in a pipeline.

3.5 Discussion

In this section, we discuss the comparison with RocketQA.

This work presents an extended contribution to RocketQA (Qu et al., 2021), called RocketQAv2. As seen from above, RocketQAv2 reuse the network architecture and important training tricks in RocketQA. A significant improvement is that RocketQAv2 incorporates a joint training approach for both the retriever and the re-ranker via dynamic listwise distillation. For dynamic listwise distillation, RocketQAv2 designs a unified listwise training approach, and utilizes soft relevance labels for mutual improvement. Such a distillation mechanism is able to simplify the training process, and also provides the possibility for end-to-end training the entire dense retrieval architecture.

4 Experiments

In this section, we first describe the experimental settings, then report the main experimental results, ablation study, and detailed analysis.

4.1 Experimental Setup

Datasets We adopt two public datasets on dense passage retrieval and passage re-ranking, including MSMARCO (Nguyen et al., 2016) and Natural Questions (Kwiatkowski et al., 2019). Table 1 lists the statistics of the datasets. *MSMARCO* was originally designed for multiple passage machine reading comprehension, and its queries were sampled from Bing search logs. Based on the queries and passages in MSMARCO Question Answering, MSMARCO Passage Ranking for passage retrieval and ranking was created. *Natural Questions (NQ)* was originally introduced for open-domain QA. This corpus consists of real queries from the Google search engine along with their long and short answer annotations from the top-ranked Wikipedia pages. DPR (Karpukhin et al., 2020) selected the queries that had short answers and processed all the Wikipedia articles as the collection of passages. In our experiments, we reuse the NQ version created by DPR.

Evaluation Metrics Following previous work, we adopt Mean Reciprocal Rank (MRR) and Recall at top k ranks (Recall@k) to evaluate the performance of passage retrieval. MRR calculates the averaged reciprocal of the rank at which the first positive passage is retrieved. Recall@k calculates the proportion of questions to which the top k retrieved passages contain positives.

Model Specifications Our retriever and reranker largely follow ERNIE-2.0 base (Sun et al., 2020), which is a BERT-like (Devlin et al., 2019) model with 12-layer transformers and introduces a continual pre-training framework on multiple pretrained tasks. As described in previous section, the retriever is initialized with the parameters of the dual-encoder in the first step of RocketQA, and the re-ranker is initialized with the parameters of the cross-encoder in the second step of RocketQA.

Implementation Details We conduct experi-

Methods	PLM	MSMARCO Dev			Natural Questions Test		
		MRR@10	R@50	R@1000	R@5	R@20	R@100
BM25 (anserini) (Yang et al., 2017)	-	18.7	59.2	85.7	-	59.1	73.7
doc2query (Nogueira et al., 2019c)	-	21.5	64.4	89.1	-	-	-
DeepCT (Dai and Callan, 2019)	-	24.3	69.0	91.0	-	-	-
docTTTTTquery (Nogueira et al., 2019a)	-	27.7	75.6	94.7	-	-	-
GAR (Mao et al., 2020)	-	-	-	-	-	74.4	85.3
UHD-BERT (Jang et al., 2021)	-	29.6	77.7	96.1	-	-	-
COIL (Gao et al., 2021a)	-	35.5	-	96.3	-	-	-
DPR (single) (Karpukhin et al., 2020)	BERT$_{\text{base}}$	-	-	-	-	78.4	85.4
DPR-E	ERNIE$_{\text{base}}$	32.5	82.2	97.3	68.4	80.7	87.3
ANCE (single) (Xiong et al., 2020a)	RoBERTa$_{\text{base}}$	33.0	-	95.9	-	81.9	87.5
TAS-Balanced (Hofstätter et al., 2021)	BERT$_{\text{base}}$	34.0	-	97.5	-	-	-
ME-BERT (Luan et al., 2021)	BERT$_{\text{large}}$	34.3	-	-	-	-	-
ColBERT (Khattab and Zaharia, 2020)	BERT$_{\text{base}}$	36.0	82.9	96.8	-	-	-
NPRINC (Lu et al., 2020)	BERT$_{\text{base}}$	31.1	-	97.7	73.3	82.8	88.4
ADORE+STAR (Zhan et al., 2021)	RoBERTa$_{\text{base}}$	34.7	-	-	-	-	-
RocketQA (Qu et al., 2021)	ERNIE$_{\text{base}}$	37.0	85.5	97.9	74.0	82.7	88.5
PAIR (Ren et al., 2021)	ERNIE$_{\text{base}}$	37.9	**86.4**	**98.2**	74.9	83.5	**89.1**
RocketQAv2 (retriever)	ERNIE$_{\text{base}}$	**38.8**	86.2	98.1	**75.1**	**83.7**	89.0

Table 2: Passage retrieval results on MSMARCO and Natural Questions datasets. PLM is the abbreviation of Pre-trained Language Model. We copy the results from original papers and we leave it blank if the original paper does not report the result. The best and second-best results are in bold and underlined fonts respectively.

ments with the deep learning framework PaddlePaddle (Ma et al., 2019) on up to 32 NVIDIA Tesla V100 GPUs (with 32G RAM). For both two datasets, we used the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 1e-5. The model is trained up to 3 epochs with a batch size of 96. The dropout rates are set to 0.1 on the cross-encoder. The ratio of the positive to the hard negative is set to 1:127 on MSMARCO and 1:31 on NQ.

4.2 Results on Passage Retrieval

In this part, we first describe the comparing baselines, then report the results on passage retrieval.

4.2.1 Baselines

To have comprehensive comparison, we choose as baselines the state-of-the-art approaches that consider both sparse and dense passage retrievers.

The sparse retrievers include the traditional retriever BM25 (Yang et al., 2017) and five traditional retrievers enhanced by neural networks, including doc2query (Nogueira et al., 2019c), DeepCT (Dai and Callan, 2019), docTTTTTquery (Nogueira et al., 2019a), GAR (Mao et al., 2020), UHD-BERT (Jang et al., 2021) and COIL (Gao et al., 2021a). Both doc2query and docTTTTTquery employ neural query generation to expand documents. In contrast, GAR employs neural generation models to expand queries and UHD-BERT is empowered by extremely high dimensionality and controllable sparsity. Different from them, DeepCT and COIL utilizes BERT to learn the term weight or inverted list.

The dense retrievers include DPR (Karpukhin et al., 2020), DPR-E, ANCE (Xiong et al., 2020a), ME-BERT (Luan et al., 2021), NPRINC (Lu et al., 2020), ColBERT (Khattab and Zaharia, 2020) RocketQA (Qu et al., 2021), TAS-Balanced (Hofstätter et al., 2021), ADORE+STAR (Zhan et al., 2021) and PAIR (Ren et al., 2021). DPR-E is our implementation of DPR using ERNIE (Sun et al., 2020) instead of BERT, which is to examine the effects of pre-trained language models.

4.2.2 Results

The results of different passage retrieval methods are presented in Table 2. It can be observed that:

(1) Among all methods, we can see the RocketQAv2 retriever and PAIR outperform other baselines by a large margin. PAIR is a contemporaneous work with RocketQAv2, which obtains improvement by pre-training on out-of-domain data. We observe that RocketQAv2 outperforms PAIR in the metrics of MRR@10 and Recall@5, we consider that dynamic listwise distillation enables the retriever to capture the re-ranker ability of passage ranking at top ranks. Our model is trained with complete in-domain training data. Different from the baselines, we adopt a listwise training ap-

Methods	**PLM**	**#candidate**	**Retriever**	**MRR@10**
BM25 (anserini) (Yang et al., 2017)	-	-	-	18.7
ColBERT (Khattab and Zaharia, 2020)	BERT$_{\text{base}}$	1000	BM25	34.9
BERT$_{\text{large}}$ (Nogueira and Cho, 2019)	BERT$_{\text{large}}$	1000	BM25	36.5
RepBERT (Zhan et al., 2020)	BERT$_{\text{large}}$	1000	RepBERT	37.7
Multi-stage (Nogueira et al., 2019b)	BERT$_{\text{base}}$	1000	BM25	39.0
CAKD (Hofstätter et al., 2020)	DistilBERT	1000	BM25	39.0
ME-BERT (Luan et al., 2021)	BERT$_{\text{large}}$	1000	ME-BERT	39.5
ME-HYBIRD (Luan et al., 2021)	BERT$_{\text{large}}$	1000	ME-HYBIRD	39.4
TFR-BERT (Han et al., 2020)	BERT$_{\text{large}}$	1000	BM25	40.5
RocketQA (Qu et al., 2021)	ERNIE$_{\text{base}}$	50	RocketQA	40.9
RocketQAv2 (re-ranker)	ERNIE$_{\text{base}}$	1000	BM25	40.1
	ERNIE$_{\text{base}}$	50	RocketQA	41.8
	ERNIE$_{\text{base}}$	50	RocketQAv2 (retriever)	**41.9**

Table 3: The MRR@10 results of different methods for passage re-ranking on MSMARCO dataset. We copy the baseline results from original papers and report the PLM, candidate number and retriever for each method.

proach to jointly train both retriever and re-ranker and couple the two models by dynamic listwise distillation with hybrid data augmentation.

(2) We notice that different PLMs are used in different approaches, as shown in the second column of Table 2. In our approach, we use ERNIE base as the backbone model. We replacing BERT base used in DPR with ERNIE base to examine the effect of the backbone model, namely DPR-E. we observe that although both two methods employ the same backbone PLM, our method significantly outperforms DPR-E, indicating that PLM is not the factor for improvement.

(3) Among sparse retrievers, we find that COIL outperforms other methods, which seems to be a robust sparse baseline that gives substantial performance on the two datasets. We also observed that sparse retrievers overall perform worse than dense retrievers, such a finding has also been reported in prior studies (Xiong et al., 2020a; Luan et al., 2021; Qu et al., 2021), which indicates the effectiveness of the dense retrieval approach.

4.3 Results on Passage Re-ranking

In this part, we first describe the comparing baselines, then report the results on passage re-ranking.

4.3.1 Baselines

We report the results of the following baselines: BM25 (Yang et al., 2017), ColBERT (Khattab and Zaharia, 2020), BERT$_{\text{large}}$ (Nogueira and Cho, 2019), RepBERT (Zhan et al., 2020), Multi-stage (Nogueira et al., 2019b), CAKD (Hofstätter et al., 2020), ME-BERT (Luan et al., 2021), ME-HYBIRD (Luan et al., 2021), TFR-BERT (Han et al., 2020) and RocketQA (Qu et al., 2021). Among these methods, BM25 is a term-based method, and the rest are BERT-based methods based on neural networks. Since RocketQA does not report re-ranking results, we use the open-source re-ranker in RocketQA repository for evaluation. We report the results of RocketQAv2 re-ranker based on BM25 retriever with 1000 candidates, RocketQA retriever with 50 candidates and RocketQAv2 retriever with 50 candidates for comparing.

The prior works follow the two-stage approach (i.e., retrieve-then-rerank), where a passage retriever retrieves a (usually large) list of candidates from the passage collection in the first stage. In the second stage, a more expensive model (e.g., BERT-based cross-encoder) re-ranks the candidates. Note that the retrievers in baseline models may be differently designed.

4.3.2 Results

Table 3 summarizes the passage re-ranking performance of RocketQAv2 re-ranker and all baselines on MSMARCO dataset.

As we can see, the RocketQAv2 re-ranker significantly outperforms all the competitive methods, demonstrating that the re-ranker benefits from our joint learning process, which is optimized to fit the relevance distribution of the retriever with dynamic listwise distillation. Morever, if we use RocketQAv2 re-ranker to replace RocketQA re-ranker and apply it on the retrieval results by RocketQA retriever, we can see that RocketQAv2 re-ranker brings 0.9 percentage point improvement comparing to RocketQA re-ranker. This also demonstrates the effectiveness of RocketQAv2 re-ranker. Additionally, if we apply RocketQAv2 re-

Methods	MRR@10	R@50
RocketQAv2 (retriever)	**37.4**	**84.9**
w/ Static Distillation	36.0	84.5
w/ Pointwise	36.3	83.9
w/o Denoised Instances	36.3	**84.9**

Table 4: The results of different variants of RocketQAv2 retriever with eight training instances per query on MSMARCO dataset. Note that the results on NQ are similar and omitted here due to limited space.

ranker on the top 1000 candidates by BM25, the performance is significantly better than other base models, and comparable to other large models.

4.4 Detailed Analysis

Apart from the above illustration, we also implement detailed analysis on both dynamic listwise distillation and hybrid data augmentation.

4.4.1 Analysis on Distillation

In this section, we analyze the results of retriever by replacing the optimization form in dynamic listwise distillation.

Dynamic or Static? To examine the effect of dynamic optimization in distillation, we utilize a well-trained cross-encoder based re-ranker as a teacher model to perform static distillation comparing with dynamic listwise distillation. During static distillation, the parameters of re-ranker model are not updated and the retriever captures the relevance knowledge from the re-ranker in a traditional knowledge distillation manner. As shown in Table 4, training with static distillation brings a performance drop. It demonstrates that dynamic optimization of both retriever and re-ranker enables to share relevance distributions with each other and brings a significant performance improvement.

Listwise or Pointwise? To study the effect of the listwise training approach, we replace it with the pointwise training approach for the re-ranker during joint training. In such case, the training approaches of the retriever and the re-ranker are actually different. The re-ranker mainly optimized by the pointwise relevance scores of instances in $\mathcal{P}_q$, while the retriever has to learn the relevance [3] from the re-ranker in a listwise way. Table 4 shows that the pointwise training approach brings performance drop, and the listwise training approach performs better in our joint training archtecture. It demonstrates that the listwise training approach is more suitable in our joint training architecture than pointwise, since it can better simulate the relevance distribution in dynamic listwise distillation.

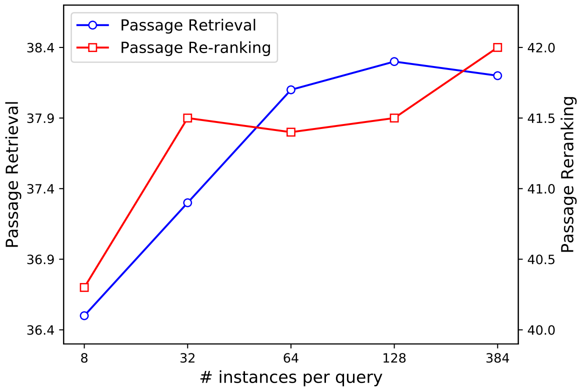

Figure 4: MRR@10 results of passage retrieval and passage re-ranking with different numbers of instances per query on MSMARCO. Note that instances per query contain one positive instance, and the rest are hard negatives.

4.4.2 Analysis on Hybrid Data Augmentation

In this section, we conduct a detailed analysis for the hybrid data augmentation.

The Effect of Denoised Instances In order to examine the effect of hybrid training data, we remove the denoised instances in training data and only use the undenoised data for joint training. Table 4 shows the performance drop in terms of MRR@10 without denoised instances, which indicates that training data generated from different ways better represent the distribution of all the passages in the whole collection, and improve the performance especially on the metrics at top ranks.

The Number of Hard Negatives In hybrid data augmentation, we focus on obtaining diverse hard negatives. In our experiments, we find that the number of hard negatives significantly affects the performance of our joint training approach. As we described in previous section, for each query, we sample one positive instance and the rest of instances in the instance list $\mathcal{P}_q$ are hard negatives. Thus, the effect of the number of hard negatives should be equivalent to the effect of the number of instances. Figure 4 shows the effect of the num-

[3] To enable learning the retriever, the pointwise relevance score from the re-ranker should be normalized in a listwise way.

ber of instances on both the passage retrieval and the passage re-ranking. From Figure 4, we can observe that a larger number of instances (*i.e.,* number of hard negatives) improves the performance. The result demonstrates that instance list $\mathcal{P}_q$ with more instances can better represent the distribution of all the passages in the whole collection.

Incorporation of In-batch Negatives For further study, we examine the effect of in-batch negatives in joint training process. Besides the hard negatives, we incorporate in-batch sampling during the joint training process, which can increase the amount of negatives for each query. Although the queries have additional in-batch negatives, we did not observe the performance improvements.

5 Conclusion

This paper has presented a novel joint training approach for dense passage retrieval and passage re-ranking. To implement the joint training, we have made two important technical contributions, namely dynamic listwise distillation and hybrid data augmentation. Such an approach is able to enhance the mutual improvement between the retriever and the re-ranker, which can also simplify the training process. Extensive results have demonstrated the effectiveness of our approach. To our knowledge, it is the first time that the retriever and re-ranker are jointly trained in a unified architecture, which provides the possibility of training the entire retrieval architecture in an end-to-end way.

Acknowledgements

This work is partially supported by the National Key Research and Development Project of China (No.2018AAA0101900), National Natural Science Foundation of China under Grant No. 61872369, Beijing Outstanding Young Scientist Program under Grant No. BJJWZYJH012019100020098 and Public Computing Cloud, Renmin University of China.

References

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, volume 227 of *ACM International Conference Proceeding Series*, pages 129–136.

Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020.*

Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 985–988.

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.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2020. Modularized transfomer-based ranking framework. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4180–4190.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021a. 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*, pages 3030–3042.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021b. Rethink training of BERT rerankers in multi-stage retrieval pipeline. In *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part II*, volume 12657, pages 280–286.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego García-Olano. 2019. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning, CoNLL 2019, Hong Kong, China, November 3-4, 2019*, pages 528–537.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: retrieval-augmented language model pre-training. *CoRR*, abs/2002.08909.

Shuguang Han, Xuanhui Wang, Mike Bendersky, and Marc Najork. 2020. Learning-to-rank with BERT in tf-ranking. *CoRR*, abs/2004.08476.

Matthew L. Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017.

Efficient natural language response suggestion for smart reply. *CoRR*, abs/1705.00652.

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.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. *CoRR*, abs/2104.06967.

Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 2553–2561.

Gautier Izacard and Edouard Grave. 2020. Distilling knowledge from reader to retriever for question answering. *CoRR*, abs/2012.04584.

Kyoungrok Jang, Junmo Kang, Giwon Hong, Sung-Hyon Myaeng, Joohee Park, Taewon Yoon, and Hee-Cheol Seo. 2021. UHD-BERT: bucketed ultra-high dimensional sparse representations for full ranking. *CoRR*, abs/2104.07198.

Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. An information retrieval approach to short text conversation. *CoRR*, abs/1408.6988.

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.

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.

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*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6086–6096.

Jing Lu, Gustavo Hernández Ábrego, Ji Ma, Jianmo Ni, and Yinfei Yang. 2020. Neural passage retrieval with improved negative contrast. *CoRR*, abs/2010.12523.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

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):105–115.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2020. Generation-augmented retrieval for open-domain question answering. *CoRR*, abs/2009.08553.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Reader-guided passage reranking for open-domain question answering. *CoRR*, abs/2101.00294.

Donald Metzler, Yi Tay, Dara Bahri, and Marc Najork. 2021. Rethinking search: Making experts out of dilettantes. *CoRR*, abs/2105.02274.

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.

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

Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019a. From doc2query to docttttquery. *Online preprint*.

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

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019c. Document expansion by query prediction. *CoRR*, abs/1904.08375.

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*, pages 5835–5847.

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

Devendra Singh Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L. Hamilton, and Bryan Catanzaro. 2021. End-to-end training of neural retrievers for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6648–6662.

Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A continual pre-training framework for language understanding. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8968–8975.

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage BERT: A globally normalized BERT model for open-domain question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5877–5881.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6397–6407.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020a. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *CoRR*, abs/2007.00808.

Wenhan Xiong, Xiang Lorraine Li, Srinivasan Iyer, Jingfei Du, Patrick S. H. Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oguz. 2020b. Answering complex open-domain questions with multi-hop dense retrieval. *CoRR*, abs/2009.12756.

Ming Yan, Chenliang Li, Chen Wu, Bin Bi, Wei Wang, Jiangnan Xia, and Luo Si. 2019. IDST at TREC 2019 deep learning track: Deep cascade ranking with generation-based document expansion and pre-trained language modeling. In *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019, Gaithersburg, Maryland, USA, November 13-15, 2019*, volume 1250 of *NIST Special Publication*.

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.

Sohee Yang and Minjoon Seo. 2020. Is retriever merely an approximator of reader? *arXiv preprint arXiv:2010.10999*.

Yinfei Yang, Ning Jin, Kuo Lin, Mandy Guo, and Daniel Cer. 2020. Neural retrieval for question answering with cross-attention supervised data augmentation. *CoRR*, abs/2009.13815.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. *CoRR*, abs/2104.08051.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. Repbert: Contextualized text embeddings for first-stage retrieval. *CoRR*, abs/2006.15498.