RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering

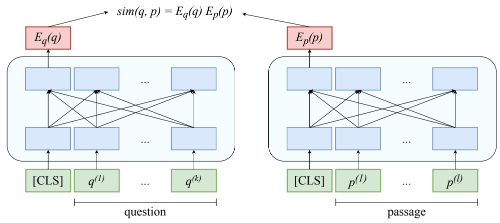
Yingqi Qu[1], Yuchen Ding[1], Jing Liu[1]*, Kai Liu[1], Ruiyang Ren[2]†
Wayne Xin Zhao[2]*, Daxiang Dong[1], Hua Wu[1] and Haifeng Wang[1]
[1]Baidu Inc.; [2]Gaoling School of Artificial Intelligence, Renmin University of China
{quyingqi, dingyuchen, liujing46, liukai20, dongdaxiang, wu_hua, wanghaifeng}@baidu.com
reyon.ren@ruc.edu.cn, batmanfly@gmail.com

Abstract

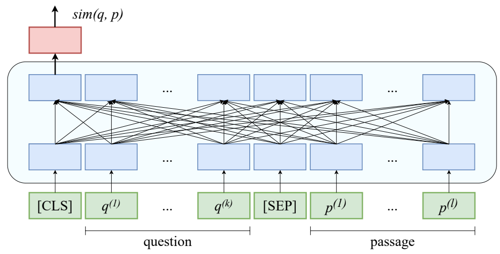
In open-domain question answering, dense passage retrieval has become a new paradigm to retrieve relevant passages for finding answers. Typically, the dual-encoder architecture is adopted to learn dense representations of questions and passages for semantic matching. However, it is difficult to effectively train a dual-encoder due to the challenges including the discrepancy between training and inference, the existence of unlabeled positives and limited training data. To address these challenges, we propose an optimized training approach, called *RocketQA*, to improving dense passage retrieval. We make three major technical contributions in RocketQA, namely cross-batch negatives, denoised hard negatives and data augmentation. The experiment results show that RocketQA significantly outperforms previous state-of-the-art models on both MS-MARCO and Natural Questions. We also conduct extensive experiments to examine the effectiveness of the three strategies in RocketQA. Besides, we demonstrate that the performance of end-to-end QA can be improved based on our RocketQA retriever [1].

1 Introduction

Open-domain question answering (QA) aims to find the answers to natural language questions from a large collection of documents. Early QA systems (Brill et al., 2002; Dang et al., 2007; Ferrucci et al., 2010) constructed complicated pipelines consisting of multiple components, including question understanding, document retrieval, passage ranking and answer extraction. Recently, inspired by the advancements of machine reading comprehension (MRC), Chen et al. (2017) proposed a simplified two-stage approach, where a traditional IR *retriever* (e.g., TF-IDF or BM25) first selects a few relevant passages as contexts, and then a neural *reader* reads the contexts and extracts the answers. As the recall component, the first-stage retriever significantly affects the final QA performance. Though efficient with an inverted index, traditional IR retrievers with term-based sparse representations have limited capabilities in matching questions and passages, e.g., term mismatch.



(a) A dual-encoder based on pre-trained LMs.



(b) A cross-encoder based on pre-trained LMs.

Figure 1: The comparison of dual-encoder and cross-encoder architectures.

To deal with the issue of term mismatch, the dual-encoder architecture (as shown in Figure 1a) has been widely explored (Lee et al., 2019; Guu et al., 2020; Karpukhin et al., 2020; Luan et al., 2020; Xiong et al., 2020) to learn dense representations of questions and passages in an end-to-end manner, which provides better representations for semantic matching. These studies first separately encode questions and passages to obtain their dense representations, and then compute the similarity between the dense representations using similarity functions such as cosine or dot product. Typically, the dual-encoder is trained by using in-batch random negatives: for each *question-positive passage*

---

* Corresponding authors.
† The work was done when Ruiyang Ren was doing internship at Baidu.
[1]Our code is available at `https://github.com/PaddlePaddle/Research/tree/master/NLP/NAACL2021-RocketQA`

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5835–5847
June 6–11, 2021. ©2021 Association for Computational Linguistics

pair in a training batch, the positive passages for the other questions in the batch would be used as negatives. However, it is still difficult to effectively train a dual-encoder for dense passage retrieval due to the following three major challenges.

First, there exists the discrepancy between training and inference for the dual-encoder retriever. During inference, the retriever needs to identify positive (or relevant) passages for each question from a large collection containing millions of candidates. However, during training, the model is learned to estimate the probabilities of positive passages in a small candidate set for each question, due to the limited memory of a single GPU (or other device). To reduce such a discrepancy, previous work tried to design specific mechanisms for selecting a few hard negatives from the top-k retrieved candidates (Gillick et al., 2019; Wu et al., 2020; Karpukhin et al., 2020; Luan et al., 2020; Xiong et al., 2020). However, it suffers from the false negative issue due to the following challenge.

Second, there might be a large number of unlabeled positives. Usually, it is infeasible to completely annotate all the candidate passages for one question. By only examining the the top-K passages retrieved by a specific retrieval approach (e.g. BM25), the annotators are likely to miss relevant passages to a question. Taking the MSMARCO dataset (Nguyen et al., 2016) as an example, each question has only 1.1 annotated positive passages on average, while there are 8.8M passages in the whole collection. As will be shown in our experiments, we manually examine the top-retrieved passages that were not labeled as positives in the original MSMARCO dataset, and we find that 70% of them are actually positives. Hence, it is likely to bring false negatives when sampling hard negatives from the top-k retrieved passages.

Third, it is expensive to acquire large-scale training data for open-domain QA. MSMARCO and Natural Questions (Kwiatkowski et al., 2019) are two largest datasets for open-domain QA. They are created from commercial search engines, and have 516K and 300K annotated questions, respectively. However, it is still insufficient to cover all the topics of questions issued by users to search engines.

In this paper, we focus on addressing these challenges so as to effectively train a dual-encoder retriever for open-domain QA. We propose an optimized training approach, called *RocketQA*, to improving dense passage retrieval. Considering the above challenges, we make three major technical contributions in RocketQA. First, RocketQA introduces cross-batch negatives. Comparing to in-batch negatives, it increases the number of available negatives for each question during training, and alleviates the discrepancy between training and inference. Second, RocketQA introduces denoised hard negatives. It aims to remove false negatives from the top-ranked results retrieved by a retriever, and derive more reliable hard negatives. Third, RocketQA leverages large-scale unsupervised data "*labeled*" by a cross-encoder (as shown in Figure 1b) for data augmentation. Though inefficient, the cross-encoder architecture has been found to be more capable than the dual-encoder architecture in both theory and practice (Luan et al., 2020). Therefore, we utilize a cross-encoder to generate high-quality pseudo labels for unlabeled data which are used to train the dual-encoder retriever. The contributions of this paper are as follows:

- The proposed RocketQA introduces three novel training strategies to improve dense passage retrieval for open-domain QA, namely cross-batch negatives, denoised hard negatives, and data augmentation.

- The overall experiments show that our proposed RocketQA significantly outperforms previous state-of-the-art models on both MSMARCO and Natural Questions datasets.

- We conduct extensive experiments to examine the effectiveness of the above three strategies in RocketQA. Experimental results show that the three strategies are effective to improve the performance of dense passage retrieval.

- We also demonstrate that the performance of end-to-end QA can be improved based on our RocketQA retriever.

2 Related Work

Passage retrieval for open-domain QA For open-domain QA, a passage retriever is an important component to identify relevant passages for answer extraction. Traditional approaches (Chen et al., 2017) implemented term-based passage retrievers (e.g. TF-IDF and BM25), which have limited representation capabilities. Recently, researchers have utilized deep learning to improve traditional passage retrievers, including document expansions (Nogueira et al., 2019c), question expansions (Mao et al., 2020) and term weight estimation (Dai and Callan, 2019).

Different from the above term-based approaches, dense passage retrieval has been proposed to represent both questions and documents as dense vectors (i.e., embeddings), typically in a dual-encoder architecture (as shown in Figure 1a). Existing approaches can be divided into two categories: (1) self-supervised pre-training for retrieval (Lee et al., 2019; Guu et al., 2020; Chang et al., 2020) and (2) fine-tuning pre-trained language models on labeled data. Our work follows the second class of approaches, which show better performance with less cost. Although the dual-encoder architecture enables the appealing paradigm of dense retrieval, it is difficult to effectively train a retriever with such an architecture. As discussed in Section 1, it suffers from a number of challenges, including the training and inference discrepancy, a large number of unlabeled positives and limited training data. Several recent studies (Karpukhin et al., 2020; Luan et al., 2020; Chang et al., 2020; Henderson et al., 2017) tried to address the first challenge by designing complicated sampling mechanism to generate hard negatives. However, it still suffers from the issue of false negatives. The later two challenges have seldom been considered for open-domain QA.

Passage re-ranking for open-domain QA Based on the retrieved passages from a first-stage retriever, BERT-based rerankers have recently been applied to retrieval-based question answering and search-related tasks (Wang et al., 2019; Nogueira and Cho, 2019; Nogueira et al., 2019b; Yan et al., 2019), and yield substantial improvements over the traditional methods. Although effective to some extent, these rankers employ the cross-encoder architecture (as shown in Figure 1b) that is impractical to be applied to all passages in a corpus with respect to a question. The re-rankers (Khattab and Zaharia, 2020; Gao et al., 2020) with light weight interaction based on the representations of dense retrievers have been studied. However, these techniques still rely on a separate retriever which provides candidates and representations. As a comparison, we focus on developing dual-encoder based retrievers.

3 Approach

In this section, we propose an optimized training approach to dense passage retrieval for open-domain QA, namely *RocketQA*. We first introduce the background of the dual-encoder architecture, and then describe the three novel training strategies in RocketQA. Lastly, we present the whole training procedure of RocketQA.

3.1 Task Description

The task of open-domain QA is described as follows. Given a natural language question, a system is required to answer it based on a large collection of documents. Let C denote the corpus, consisting of N documents. We split the N documents into M passages, denoted by p_1, p_2, ..., p_M, where each passage p_i can be viewed as an l-length sequence of tokens $p_i^{(1)}$, $p_i^{(2)}$, ..., $p_i^{(l)}$. Given a question q, the task is to find a passage p_i among the M candidates, and extract a span $p_i^{(s)}$, $p_i^{(s+1)}$, ..., $p_i^{(e)}$ from p_i that can answer the question. In this paper, we mainly focus on developing a dense retriever to retrieve the passages that contain the answer.

3.2 The Dual-Encoder Architecture

We develop our passage retriever based on the typical dual-encoder architecture, as illustrated in Figure 1a. First, a dense passage retriever uses an encoder $E_p(\cdot)$ to obtain the d-dimensional real-valued vectors (a.k.a., embedding) of passages. Then, an index of passage embeddings is built for retrieval. At query time, another encoder $E_q(\cdot)$ is applied to embed the input question to a d-dimensional real-valued vector, and k passages whose embeddings are the closest to the question's will be retrieved. The similarity between the question q and a candidate passage p can be computed as the dot product of their vectors:

$$\text{sim}(q,p) = E_q(q) \cdot E_p(p). \tag{1}$$

In practice, the separation of question encoding and passage encoding is desirable, so that the dense representations of all passages can be precomputed for efficient retrieval. Here, we adopt two independent neural networks initialized from pre-trained LMs for the two encoders $E_q(\cdot)$ and $E_p(\cdot)$ separately, and take the representations at the first token (e.g., [CLS] symbol in BERT) as the output for encoding.

Training The training objective is to learn dense representations of questions and passages so that *question-positive passage* pairs have higher similarity than the *question-negative passage* pairs in training data. Formally, given a question q_i together with its positive passage p_i^+ and m negative passages $\{p_{i,j}^-\}_{j=1}^m$, we minimize the loss function:

$$\begin{aligned} &\mathcal{L}(q_i, p_i^+, \{p_{i,j}^-\}_{j=1}^m) \\ &= -\log \frac{e^{\text{sim}(q_i,p_i^+)}}{e^{\text{sim}(q_i,p_i^+)} + \sum_{j=1}^m e^{\text{sim}(q_i,p_{i,j}^-)}}, \end{aligned} \tag{2}$$

where we aim to optimize the negative log likelihood of the positive passage against a set of m negative passages. Ideally, we should take all the negative passages in the whole collection into consideration in Equation 2. However, it is computationally infeasible to consider a large number of negative samples for a question, and hence m is practically set to a small number that is far less than M. As what will be discussed later, both the number and the quality of negatives affect the final performance of passage retrieval.

Inference In our implementation, we use FAISS (Johnson et al., 2019) to index the dense representations of all passages. Specifically, we use IndexFlatIP for indexing and the exact maximum inner product search for querying.

3.3 Optimized Training Approach

In Section 1, we have discussed three major challenges in training the dual-encoder based retriever, including the training and inference discrepancy, the existence of unlabeled positives, and limited training data. Next, we propose three improved training strategies to address the three challenges.

Cross-batch Negatives When training the dual-encoder, the trick of in-batch negatives has been widely used in previous work (Henderson et al., 2017; Gillick et al., 2019; Wu et al., 2020; Karpukhin et al., 2020; Luan et al., 2020). Assume that there are B questions in a mini-batch on a single GPU, and each question has one positive passage. With the in-batch negative trick, each question can be further paired with $B-1$ negatives (i.e., positive passages of the rest questions) without sampling additional negatives. In-batch negative training is a memory-efficient way to reuse the examples already loaded in a mini-batch rather than sampling new negatives, which increases the number of negatives for each question. As illustrated at the top of Figure 2, we present an example for in-batch negatives when training on A GPUs in a data parallel way. To further optimize the training with more negatives, we propose to use cross-batch negatives when training on multiple GPUs, as illustrated at the bottom of Figure 2. Specifically, we first compute the passage embeddings within each single GPU, and then share these passage embeddings among all the GPUs. Besides the in-batch negatives, we collect all passages (i.e., their dense representations) from other GPUs as the additional negatives for each question. Hence, with A GPUs

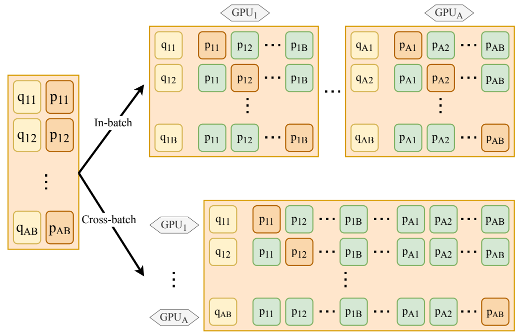


Figure 2: The comparison of traditional in-batch negatives and our cross-batch negatives when trained on multiple GPUs, where A is the number of GPUs, and B is the number of questions in each min-batch.

(or mini-batches) [2], we can indeed obtain $A \times B - 1$ negatives for a given question, which is approximately A times as many as the original number of in-batch negatives. In this way, we can use more negatives in the training objective of Equation 2, so that the results are expected to be improved.

Denoised Hard Negatives Although the above strategy can increase the number of negatives, most of negatives are easy ones, which can be easily discriminated. While, hard negatives are shown to be important to train a dual-encoder (Gillick et al., 2019; Wu et al., 2020; Karpukhin et al., 2020; Luan et al., 2020; Xiong et al., 2020). To obtain hard negatives, a straightforward method is to select the top-ranked passages (excluding the labeled positive passages) as negative samples. However, it is likely to bring false negatives (i.e., unlabeled positives), since the annotators can only annotate a few top-retrieved passages (as discussed in Section 1). Another note is that previous work mainly focuses on factoid questions, to which the answers are short and concise. Hence, it is not challenging to filter false negatives by using the short answers (Karpukhin et al., 2020). However, it cannot apply to non-factoid questions. In this paper, we aim to learn dense passage retrieval for both factoid questions and non-factoid questions, which needs a more effective way for denoising hard negatives.

Here, our idea is to utilize a well-trained cross-encoder to remove top-retrieved passages that are likely to be false negatives. Because the cross-encoder architecture is more powerful for capturing semantic similarity via deep interaction and shows much better performance than the dual-encoder ar-

[2] Note that cross-batch negatives can be applied in both settings of single-GPU and multi-GPUs. When there is only a single GPU available, it can be implemented in an accumulation way while trading off training time.

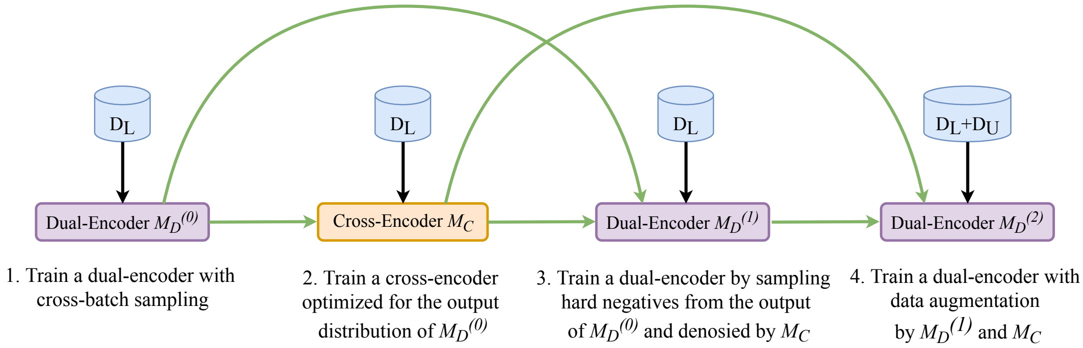


Figure 3: The pipeline of the optimized training approach RocketQA. M_D and M_C denote the dual-encoder and cross-encoder, respectively. We use $M_D^{(0)}$, $M_D^{(1)}$ and $M_D^{(2)}$ to denote the learned dual-encoders after different steps.

chitecture (Luan et al., 2020). The cross-encoder is more effective and robust, while it is inefficient over a large number of candidates in inference. Hence, we first train a cross-encoder (following the architecture shown in Figure 1b). Then, when sampling hard negatives from the top-ranked passages retrieved by a dense retriever, we select only the passages that are predicted as negatives by the cross-encoder with high confidence scores. The selected top-retrieved passages can be considered as denosied samples that are more reliable to be used as hard negatives.

Data Augmentation The third strategy aims to alleviate the issue of limited training data. Since the cross-encoder is more powerful in measuring the similarity between questions and passages, we utilize it to annotate unlabeled questions for data augmentation. Specifically, we incorporate a new collection of unlabeled questions, while reuse the passage collection. Then, we use the learned cross-encoder to predict the passage labels for the new questions. To ensure the quality of the automatically labeled data, we only select the predicted positive and negative passages with high confidence scores estimated by the cross-encoder. Finally, the automatically labeled data is used as augmented training data to learn the dual encoder. Another view of the data augmentation is knowledge distillation (Hinton et al., 2015), where the cross-encoder is the teacher and the dual-encoder is the student.

3.4 The Training Procedure

As shown in Figure 3, we organize the above three training strategies into an effective training pipeline for the dual-encoder. It makes an analogy to a multi-stage rocket, where the performance of the dual-encoder is consecutively improved at three steps (STEP 1, 3 and 4). That is why we call our approach *RocketQA*. Next, we will describe the details of the whole training procedure of RocketQA.

- **REQUIRE:** Let C denote a collection of passages. Q_L is a set of questions that have corresponding labeled passages in C, and Q_U is a set of questions that have no corresponding labeled passages. D_L is a dataset consisting of C and Q_L, and D_U is a dataset consisting of C and Q_U.
- **STEP 1:** Train a dual-encoder $M_D^{(0)}$ by using cross-batch negatives on D_L.
- **STEP 2:** Train a cross-encoder M_C on D_L. The positives used for training the cross-encoder are from the original training set D_L, while the negatives are randomly sampled from the top-k passages (excluding the labeled positive passages) retrieved by $M_D^{(0)}$ from C for each question $q \in Q_L$. This design is to let the cross-encoder adjust to the distribution of the results retrieved by the dual-encoder, since the cross-encoder will be used in the following two steps for optimizing the dual-encoder. This design is important, and there is similar observation in Facebook Search (Huang et al., 2020).
- **STEP 3:** Train a dual-encoder $M_D^{(1)}$ by further introducing denoised hard negative sampling on D_L. Regarding to each question $q \in Q_L$, the hard negatives are sampled from the top passages retrieved by $M_D^{(0)}$ from C, and only the passages that are predicted as negatives by the cross-encoder M_C with high confidence scores will be selected.
- **STEP 4:** Construct pseudo training data D_U by using M_C to label the top-k passages retrieved by $M_D^{(1)}$ from C for each question $q \in Q_U$, and then train a dual-encoder $M_D^{(2)}$ on both the manually labeled training data D_L and the automatically augmented training data D_U.

Note that the cross-batch negative strategy is applied through all the steps for training the dual-

datasets	#q in train	#q in dev	#q in test	#p	ave. q length	ave. p length
MSMARCO	502,939	6,980	6,837	8,841,823	5.97	56.58
NQ	58,812	-	3,610	21,015,324	9.20	100.0

Table 1: The statistics of datasets MSMARCO and Natural Questions. Here, "p" and "q" are the abbreviations of questions and passages, respectively. The length is in tokens.

encoder. The cross-encoder is used both STEP 3 and STEP 4 with different purposes to promote the performance of the dual encoder. The implementation details of denoising hard negatives and data augmentation can be found in Section 4.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets

We conduct the experiments on two popular QA benchmarks: MSMARCO Passage Ranking (Nguyen et al., 2016) and Natural Questions (NQ) (Kwiatkowski et al., 2019). The statistics of the datasets are listed in Table 1.

MSMARCO Passage Ranking MSMARCO is originally designed for multiple passage MRC, and its questions were sampled from Bing search logs. Based on the questions and passages in MSMARCO Question Answering, a dataset for passage ranking was created, namely MSMARCO Passage Ranking, consisting of about 8.8 million passages. The goal is to find positive passages that answer the questions.

Natural Question (NQ) Kwiatkowski et al. (2019) introduces a large dataset for open-domain QA. The original dataset contains more than $300,000$ questions collected from Google search logs. In Karpukhin et al. (2020), around $62,000$ factoid questions are selected, and all the Wikipedia articles are processed as the collection of passages. There are more than 21 million passages in the corpus. In our experiments, we reuse the version of NQ created by Karpukhin et al. (2020). Note that the dataset used in DPR contains empty negatives, and we discarded the empty ones.

4.1.2 Evaluation Metrics

Following previous work, we use MRR and Recall at top k ranks to evaluate the performance of passage retrieval, and exact match (EM) to measure the performance of answer extraction.

MRR The Reciprocal Rank (RR) calculates the reciprocal of the rank at which the first relevant passage was retrieved. When averaged across questions, it is called Mean Reciprocal Rank (MRR).

Recall at top k ranks The top-k recall of a retriever is defined as the proportion of questions to which the top k retrieved passages contain answers.

Exact match This metric measures the percentage of questions whose predicted answers that match any one of the reference answers exactly, after string normalization.

4.1.3 Implementation Details

We conduct all experiments with the deep learning framework PaddlePaddle (Ma et al., 2019) on up to eight NVIDIA Tesla V100 GPUs (with 32G RAM).

Pre-trained LMs The dual-encoder is initialized with the parameters of ERNIE 2.0 base (Sun et al., 2020), and the cross-encoder is initialized with ERNIE 2.0 large. ERNIE 2.0 has the same networks as BERT, and it introduces continual pre-training framework on multiple pre-trained tasks. We notice previous work use different pre-trained LMs, and we examine the effects of pre-trained LMs in Section A.1 in Appendix. Our approach is effective when using different pre-trained LMs.

Cross-batch negatives [3] The cross-batch negative sampling is implemented with differentiable all-gather operation provided in FleetX (Dong, 2020), that is a highly scalable distributed training engine of PaddlePaddle. The all-gather operator makes representation of passages across all GPUs visible on each GPU and thus the cross-batch negative sampling approach can be applied globally.

Denoised hard negatives and data augmentation We use the cross-encoder for both denoising hard negatives and data augmentation. Specifically, we select the top retrieved passages with scores less than 0.1 as negatives and those with scores higher than 0.9 as positives. We manually evaluated the selected data, and the accuracy was higher than 90%.

The number of positives and negatives When training the cross-encoders, the ratios of the number of positives to the number of negatives are 1:4 and 1:1 on MSMARCO and NQ, respectively. The

[3]When using multi-GPUs, the cross-batch negatives is as efficient as the in-batch negatives. Because the cross-batch re-uses the computed embeddings of paragraphs and the communication cost of embeddings across GPUs can be negligible.

Methods	PLMs	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
DPR (single) (Karpukhin et al., 2020)	$\text{BERT}_{\text{base}}$	-	-	-	-	78.4	85.4
ANCE (single) (Xiong et al., 2020)	$\text{RoBERTa}_{\text{base}}$	33.0	-	95.9	-	81.9	87.5
ME-BERT (Luan et al., 2020)	$\text{BERT}_{\text{large}}$	33.8	-	-	-	-	-
RocketQA	$\text{ERNIE}_{\text{base}}$	**37.0**	**85.5**	**97.9**	**74.0**	**82.7**	**88.5**

Table 2: The performance comparison on passage retrieval. Note that we directly copy the reported numbers from the original papers and leave the blanks if they were not reported.

negatives used for training cross-encoders are randomly sampled from top-1000 and top-100 passages retrieved by the dual-encoder $M_D^{(0)}$ on MSMARCO and NQ, respectively. When training the dual-encoders in the last two steps ($M_D^{(1)}$ and $M_D^{(2)}$), we set the ratios of the number of positives to the number of hard negatives as 1:4 and 1:1 on MSMARCO and NQ, respectively.

Batch sizes The dual-encoders are trained with the batch sizes of 512×8 and 512×2 on MSMARCO and NQ, respectively. The batch size used on MSMARCO is larger, since the size of MSMARCO is larger than NQ. The cross-encoders are trained with the batch sizes of 64×4 and 64 on MSMARCO and NQ, respectively. We use the automatic mixed precision and gradient checkpoint [4] functionality in FleetX, so as we can train the models using large batch sizes with limited resources.

Training epochs The dual-encoders are trained on MSMARCO for 40, 10 and 10 epochs in three steps of RocketQA, respectively. The dual-encoders are trained on NQ for 30 epochs in all steps of RocketQA. The cross-encoders are trained for 2 epochs on both MSMARCO and NQ.

Optimizers We use ADAM optimizer.

Warmup and learning rate The learning rate of the dual-encoder is set to 3e-5 and the rate of linear scheduling warm-up is set to 0.1, while the learning rate of the cross-encoder is set to 1e-5.

Maximal length We set the maximal length of questions and passages as 32 and 128, respectively.

Unlabeled questions We collect 1.7 million unlabeled questions from Yahoo! Answers[5], ORCAS (Craswell et al., 2020) and MRQA (Fisch et al., 2019). We use the questions from Yahoo! Answers, ORCAS and NQ as new questions in the experiments of MSMARCO. We only use the questions from MRQA as the new questions in the experiments of NQ. Since both NQ and MRQA mainly contain factoid-questions, while other datasets contain both factoid and non-factoid questions.

[4]The gradient checkpoint (Chen et al., 2016) enables the trading off computation against memory resulting in sublinear memory cost, so bigger/deeper nets can be trained with limited resources.

[5]http://answers.yahoo.com/

4.2 Experimental Results

In our experiments, we first examine the effectiveness of our retriever on MSMARCO and NQ datasets. Then, we conduct extensive experiments to examine the effects of the three proposed training strategies. We also show the performance of end-to-end QA based on our retriever on NQ dataset.

4.2.1 Dense Passage Retrieval

We first compare RocketQA with the previous state-of-the-art approaches on passage retrieval. We consider both sparse and dense passage retriever baselines. The sparse retrievers include the traditional retriever BM25 (Yang et al., 2017), and four traditional retrievers enhanced by neural networks, including doc2query (Nogueira et al., 2019c), DeepCT (Dai and Callan, 2019), docTTTTTquery (Nogueira et al., 2019a) and GAR (Mao et al., 2020). Both doc2query and docTTTTTquery employ neural question generation to expand documents. In contrast, GAR employs neural generation models to expand questions. Different from them, DeepCT utilizes BERT to learn the term weight. The dense passage retrievers include DPR (Karpukhin et al., 2020), ME-BERT (Luan et al., 2020) and ANCE (Xiong et al., 2020). Both DRP and ME-BERT use in-batch random sampling and hard negative sampling from the results retrieved by BM25, while ANCE enhances the hard negative sampling by using the dense retriever.

Table 2 shows the main experimental results. We can see that RocketQA significantly outperforms all the baselines on both MSMARCO and

Strategy	MRR@10
In-batch negatives	32.39
Cross-batch negatives (i.e. STEP 1)	33.32
Hard negatives w/o denoising	26.03
Hard negatives w/ denoising (i.e. STEP 3)	36.38
Data augmentation (i.e. STEP 4)	**37.02**

Table 3: The experiments to examine the effectiveness of the three proposed training strategies in RocketQA on MSMARCO Passage Ranking.

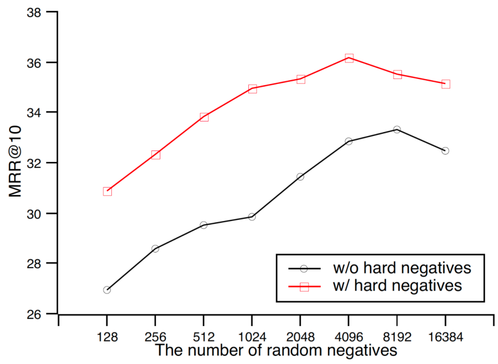


Figure 4: The effect of the number of random negatives paired for a question on MSMARCO dataset. The models without and with hard negatives are trained with 20K and 5K steps, respectively.

NQ datasets. Another observation is that the dense retrievers are overall better than the sparse retrievers. Such a finding has also been reported in previous studies (Karpukhin et al., 2020; Luan et al., 2020; Xiong et al., 2020), which indicates the effectiveness of the dense retrieval approach.

4.2.2 The Effectiveness of The Three Training Strategies in RocketQA

In this part, we conduct the extensive experiments on MSMARCO dataset to examine the effectiveness of the three strategies in RocketQA. Results on NQ dataset has shown the similar findings (see in Section A.2 in Appendix).

First, we compare cross-batch negatives with in-batch negatives by using the same experimental setting (i.e. the number of epochs is 40 and the batch size is 512 on each single GPU). From the first two rows in Table 3, we can see that the performance of the dense retriever can be improved with more negatives by cross-batch negatives. It is expected that when increasing the number of random negatives, it will reduce the discrepancy between training and inference. Furthermore, we investigate the effect of the number of random negatives. Specifically, we examine the performance of dual-encoders trained by using different numbers of random negatives with a fixed number of steps. From Figure 4, we can see that the model performance increases, when the number of random negatives becomes larger. After a certain point, the model performance starts to drop, since a large batch size may bring difficulty for optimization on training data with limited size. We say that there should be a balance between the batch size and the number of negatives. When increasing the batch size, we will have more negatives for each question. However, when the size of training data is limited, a large batch size will bring difficulty for optimization.

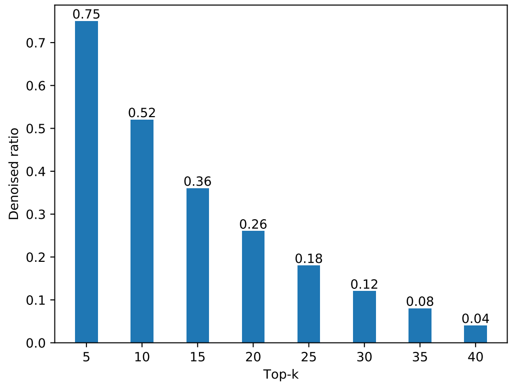


Figure 5: The ratios of denoised passages at different ranks on MSMARCO.

Second, we examine the effect of denoised hard negatives from the top-k passages retrieved by the dense retriever. As shown in the third row in Table 3, the performance of the retriever significantly decreases by introducing hard negatives without denoising. We speculate that it is caused by the fact that there are a large number of unlabeled positives. Specifically, we manually examine the top-retrieved passages of 100 questions, that were not labeled as true positives. We find that about 70% of them are actually positives or highly relevant. Hence, it is likely to bring noise if we simply sample hard negatives from the top-retrieved passages by the dense retriever, which is a widely adopted strategy to sample hard negatives in previous studies (Gillick et al., 2019; Wu et al., 2020; Xiong et al., 2020). As a comparison, we propose denoised hard negatives by a powerful cross-encoder. From the fourth row in Table 3, we can see that denoised negatives improve the performance of the dense retriever. To obtain more insights about denoised hard negatives, Table 4 gives the sampled hard negatives for two questions before and after denoising. Figure 5 further illustrates the ratio of filtered passages at different ranks. We can see that there are more passages filtered (i.e. denoised) at

Question	Label positives	Hard negatives w/o denoising (false negatives)	Hard negatives w/ denoising
How many kilohertz in a megahertz	One **megahertz (abbreviated: MHz) is equal to 1,000 kilohertz, or 1,000,000 hertz**. It can also be described as one million cycles per second. ...	(*Rank 2nd*) Kilo means times 1000, mega means times 1,000,000. So **0.005 megahertz** = 5000 Hz = **5 kiloHz**. Hertz (not Herz) is abbreviated to Hz. ...	(*Rank 14th*) ...**megahertz (MHz) and gigahertz (GHz)** are used to measure CPU speed. For example, a 1.6 GHz computer processes data internally ...
Name of test for achilles tendon rupture	In a patient with a **ruptured Achilles tendon**, the foot will not move. That is called a positive **Thompson test**. The Thompson test is important because...	(*Rank 1st*) ...The physical examination should include two or more of the following tests to establish the diagnosis of acute **Achilles tendon rupture**: Clinical **Thompson test** ...	(*Rank 9th*)...Methods: Ultrasound was used to measure **Achilles tendon**. length and muscle-tendon architectural parameters in children. of ages 5 to 12 years. ...

Table 4: The hard negatives before and after denoising on MSMARCO. The bolded words are the keywords relevant to questions.

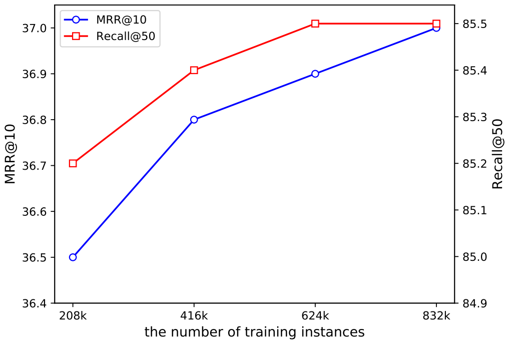


Figure 6: The effect of the size of augmented data on MSMARCO dataset.

Model	EM
BM25+BERT (Lee et al., 2019)	26.5
HardEM (Min et al., 2019a)	28.1
GraphRetriever (Min et al., 2019b)	34.5
PathRetriever (Asai et al., 2020)	32.6
ORQA (Lee et al., 2019)	33.3
REALM (Guu et al., 2020)	40.4
DPR (Karpukhin et al., 2020)	41.5
GAR (Mao et al., 2020)	41.6
RocketQA + DPR reader	42.0
RocketQA + re-trained DPR reader	**42.8**

Table 5: The experimental results of passage reading on NQ dataset. In this paper, we focus on extractive reader, while the recent generative readers (Lewis et al., 2020; Izacard and Grave, 2020) can also be applied here and may lead to better results.

lower ranks, since it is likely to have more false negatives at lower ranks.

Finally, when integrated with the data augmentation strategy (see the fifth row in Table 3), the performance has been further improved. A major merit of data augmentation is that it does not explicitly rely on manually-labeled data. Instead, it utilizes the cross-encoder (having more powerful capability than the dual-encoder) to generate pseudo training data for improving the dual-encoder. We further examine the effect of the size of the augmented data. As shown in Figure 6, we can see when the size of the augmented data is increasing, the performance increases.

4.2.3 Passage Reading with RocketQA

Previous experiments have shown the effectiveness of RocketQA on passage retrieval. Next, we verify whether the retrieval results of RocketQA can improve the performance of passage reading for extracting correct answers. We implement an end-to-end QA system in which we have an extractive reader stacked on our RocketQA retriever. For a fair comparison, we first re-use the released model [6] of the extractive reader in DPR (Karpukhin et al., 2020), and take 100 retrieved passages during inference (the same setting used in DPR). Besides, we use the same setting to train a new extractive reader based on the retrieval results of RocketQA (except that we choose top 50 passages for training instead of 100). The motivation is that the reader should be adapted to the retrieval distribution of RocketQA.

Table 5 summarizes the the end-to-end QA performance of our approach and a number of competitive methods. From Table 5, we can see that our retriever leads to better QA performance. Compared with prior solutions, our novelty mainly lies in the passage retrieval component, i.e., the RocketQA approach. The results have shown that our approach can provide better passage retrieval results, which finally improve the final QA performance.

5 Conclusions

In this paper, we have presented an optimized training approach to improving dense passage retrieval. We have made three major technical contributions in RocketQA, namely cross-batch negatives, denoised hard negatives and data augmentation. Extensive experiments have shown the effectiveness of the proposed approach by incorporating the three optimization strategies. We also demonstrate that the performance of end-to-end QA can be improved based on our RocketQA retriever.

[6] https://github.com/facebookresearch/DPR

6 Ethical Considerations

The technique of dense passage retrieval is effective for question answering, where the majority of questions are informational queries. Different from the traditional search, there is usually term mismatch between questions and answers. The term mismatch brings barriers for the machine to accurately find the information for people. Hence, we need dense passage retrieval for semantic matching in the scenario of question answering. Dense passage retrieval has the potential to empower people to find the accurate information more quickly and achieve more in their daily life and work. Our technique contributes toward the goal of asking machines to find the answers to natural language questions from a large collection of documents. However, the goal is still far from being achieved, and more efforts from the community is needed for us to get there.

7 Acknowledgments

This work is supported by the National Key Research and Development Project of China (No. 2018AAA0101900). We would also like to thank the anonymous reviewers for their insightful suggestions.

References

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Eric Brill, Susan T. Dumais, and Michele Banko. 2002. An analysis of the askmsr question-answering system. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, EMNLP 2002, Philadelphia, PA, USA, July 6-7, 2002*, pages 257–264.

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*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1870–1879.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. 2016. Training deep nets with sublinear memory cost. *CoRR*, abs/1604.06174.

Nick Craswell, Daniel Campos, Bhaskar Mitra, Emine Yilmaz, and Bodo Billerbeck. 2020. ORCAS: 20 million clicked query-document pairs for analyzing search. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 2983–2989.

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.

Hoa Trang Dang, Diane Kelly, and Jimmy J. Lin. 2007. Overview of the TREC 2007 question answering track. In *Proceedings of The Sixteenth Text REtrieval Conference, TREC 2007, Gaithersburg, Maryland, USA, November 5-9, 2007*, volume 500-274 of *NIST Special Publication*.

Daxiang Dong. 2020. paddle.distributed.fleet: A highly scalable distributed training engine of paddlepaddle.

David A. Ferrucci, Eric W. Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John M. Prager, Nico Schlaefer, and Christopher A. Welty. 2010. Building watson: An overview of the deepqa project. *AI Mag.*, 31(3):59–79.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering, MRQA@EMNLP 2019, Hong Kong, China, November 4, 2019*, pages 1–13.

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. Association for Computational Linguistics.

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.

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.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

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. Leveraging passage retrieval with generative models for open domain question answering. *CoRR*, abs/2007.01282.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

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.

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. Association for Computational Linguistics.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2020. Sparse, dense, and attentional representations for text retrieval. *CoRR*, abs/2005.00181.

Y. Ma, D. Yu, T. Wu, and H. Wang. 2019. Paddlepaddle: An open-source deep learning platform from industrial practice.

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.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019a. A discrete hard EM approach for weakly supervised 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 2851–2864.

Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019b. Knowledge guided text retrieval and reading for open domain question answering. *CoRR*, abs/1911.03868.

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*.

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.

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. AAAI Press.

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. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *CoRR*, abs/2007.00808.

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.

A Appendix

A.1 The Effects of Pre-trained LMs

We notice that previous work use different pre-trained LMs. As shown in Table 6, DPR (Karpukhin et al., 2020) uses $\text{BERT}_{\text{base}}$. ANCE (Xiong et al., 2020) uses $\text{RoBERTa}_{\text{base}}$, and ME-BERT (Luan et al., 2020) uses $\text{BERT}_{\text{large}}$. We mainly use $\text{ERNIE}_{\text{base}}$ in our experiments. In this section, we try to examine the effects of pre-trained LMs for RocketQA. Specifically, we use $\text{BERT}_{\text{base}}$ to replace $\text{ERNIE}_{\text{base}}$, and apply it to the first step of RocketQA. From Table 6 (see the forth row and the fifth row), we can observe that the performance slightly decreases when using $\text{BERT}_{\text{base}}$. In other words, comparing to $\text{BERT}_{\text{base}}$, $\text{ERNIE}_{\text{base}}$ brings gains about 0.6 in terms of MRR@10 on MSMARCO, and 1.6 in terms of R@100 on NQ, respectively. However, RocketQA trained only with cross-batch negatives is already comparable to previous work, including DPR, ANCE and ME-BERT (although they employ better pre-trained LMs). We conclude that our approach is still effective when using different pre-trained LMs.

Methods	**PLMs**	**MSMARCO** MRR@10	**NQ** R@100
DPR (single)	$\text{BERT}_{\text{base}}$	-	85.4
ANCE (single)	$\text{RoBERTa}_{\text{base}}$	33.0	87.5
ME-BERT	$\text{BERT}_{\text{large}}$	33.8	-
$\text{RocketQA}_{\text{STEP1}}$	$\text{BERT}_{\text{base}}$	32.7	86.0
$\text{RocketQA}_{\text{STEP1}}$	$\text{ERNIE}_{\text{base}}$	33.3	87.6
RocketQA	$\text{ERNIE}_{\text{base}}$	37.0	88.5

Table 6: The effects of pre-trained LMs. Note that we directly copy the reported numbers from the original papers and leave the blanks if they were not reported.

A.2 The Effectiveness of The Three Training Strategies on NQ

In this section, we examine the effectiveness of the three proposed training strategies on NQ dataset. From Table 7, we can observe that all the three strategies are effective. The findings are similar to the results on MSMARCO.

Strategy	**R@5**
In-batch negatives	68.5
Cross-batch negatives (i.e. STEP 1)	68.9
Hard negatives w/o denoising	68.0
Hard negatives w/ denoising (i.e. STEP 3)	73.2
Data augmentation (i.e. STEP 4)	**74.0**

Table 7: The experiments to examine the effectiveness of the three proposed training strategies in RocketQA on NQ.