稠密

# 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], Xin Zhao[2], Daxiang Dong[1], Hua Wu[1] and Haifeng Wang[1]**
[1]Baidu Inc., [2]Renmin University

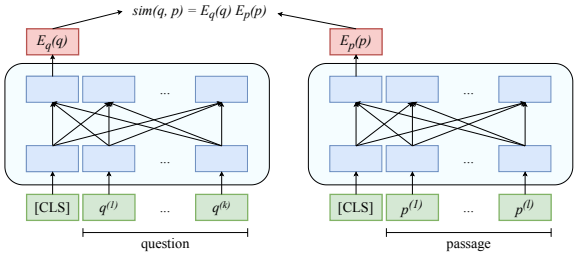
## Abstract

In open-domain question answering, dense passage retrieval has become a new paradigm to retrieve relevant passages for answer finding. Typically, the dual-encoder architecture is adopted to learn dense representations of questions and passages for matching. However, it is difficult to train an effective 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 negative sampling and data augmentation. Extensive experiments show that RocketQA significantly outperforms previous state-of-the-art models on both MS-MARCO and Natural Questions. Besides, built upon RocketQA, we achieve the first rank at the leaderboard of MSMARCO Passage Ranking Task.

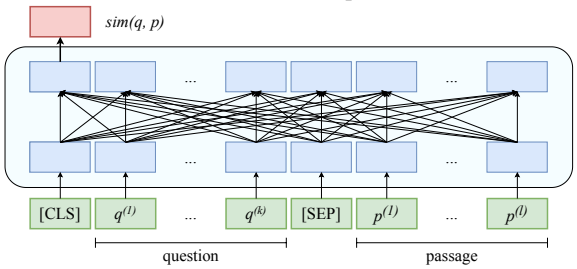
## 1 Introduction

Open-domain question answering (QA) aims to find the answers to questions expressed in natural language 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.

In recent years, 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 the support of 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 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; Chang et al., 2020; Henderson et al., 2017) to learn dense representations of questions and passages in an end-to-end manner, which provides a more powerful representation way for semantic match. These studies first separately encode questions and passages, 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* 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 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 optimize the probabilities of positive passages in a small candidate set for each question, due to the limited memory of a single GPU. To reduce such a discrepancy, previous work tried to design specific mechanisms for selecting hard negatives (Gillick et al., 2019; Wu et al., 2019; 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 from a collection of 8.8M passages. As will be shown in our experiments, we manually examine the top-ranked passages (retrieved by our retriever) that were not labeled as positives in the original MSMARCO dataset, and we find that 70% of them are actually positives or highly relevant. Hence, it is likely to contain false negatives when sampling hard negatives from the top ranked 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 public datasets to date for open-domain QA obtained from commercial search engines, which have 516K and 62K [1] annotated questions, respectively. However, it is still insufficient to cover all the topics from a variety of domains, much smaller than the number of possible questions issued by users to search engines [2].

[1] In our experiments, we use the open version of NQ created by Karpukhin et al. (2020).

[2] https://seotribunal.com/blog/google-stats-and-facts/

In this paper, we focus on addressing these challenges so as to effectively train the 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 random negatives. Comparing to in-batch negatives, it increases the number of available negatives for each question when training, and alleviates the discrepancy between training and inference. Second, RocketQA introduces denoised hard negative sampling. It aims to remove false negatives from the top-ranked results retrieved by a specific retriever, which can derive more reliable hard negatives. Third, RocketQA leverages large-scale unsupervised data "*labeled*" by a (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 for training the dual-encoder retriever.

In summary, the contributions of this paper are as follows:

- The proposed RocketQA introduces three novel optimization strategies for improving dense passage retrieval for open-domain QA, namely cross-batch sampling, denoised hard negative sampling, 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 all the three strategies are effective to improve the performance of dense passage retrieval.
- We also demonstrate that RocketQA leads to better end-to-end QA performance. Besides, built upon RocketQA, we achieve the first rank at the leaderboard of MSMARCO Passage Ranking Task.

## 2 Related Work

In this section, we review the related work in two aspects.

### 2.1 Passage Retrieval for Open-domain QA

For open-domain QA, passage retriever is an important component to identify relevant passages for answer extraction. Traditional approaches (Chen et al., 2017) implemented term-based passage retriever (e.g. TF-IDF and BM25), which have limited representation capabilities. Recently, researchers have utilized deep learning to improve traditional passage retriever. Nogueira et al. (2019c) first enriched document content by generating related queries and appending them to the documents, and then indexed the expanded documents for retrieval, which alleviate the term mismatch between questions and passages. Mao et al. (2020) used generation model to expand questions. DeepCT (Dai and Callan, 2019) utilized BERT to learn the term importance (i.e. term weighting) by considering the contexts, which was shown to achieve better retrieval performance than BM25.

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 neural architecture (as shown in Figure 1a). Existing approaches can be roughly divided into two categories: (1) pre-training and (2) fine-tuning only. Lee et al. (2019) proposed a specific approach to pre-training the retriever with an unsupervised task, namely Inverse Cloze Task (ICT), and then jointly fine-tuning the retriever and the reader on labeled data. Guu et al. (2020) proposed a new pre-training approach namely REALM, which jointly trained a masked language model and a neural retriever. In contrast, the second class of approaches only fine-tuned (existing) pre-trained language models (LMs) 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 such a retrieval 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 samples. However, it still suffers from the issue of false negatives. The later two challenges have seldom been considered for open-domain QA. Considering these challenges, it requires a systematic study on how to optimize the dual-encoder architecture for passage retrieval.

### 2.2 Passage Re-ranking for Open-domain QA

Based on the retrieved passages from a first-stage traditional retriever, a number of (pre-BERT) neural models, such as DRMM (Guo et al., 2016), KNRM (Xiong et al., 2017), and DUET (Mitra et al., 2017), have been proposed to re-rank the top-$k$ candidates. However, it has been shown (Lin, 2019) that these neural ranking models do not yield substantial improvement over traditional methods due to the lack of large-scale labeled retrieval data.

More recently, Microsoft has released MS MARCO (Nguyen et al., 2016) that is a large dataset for open-domain QA and passage re-ranking task, to foster the research along this line. Besides, the pre-trained LMs, such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018) have achieved impressive results on a variety of NLP tasks. Specially, BERT has 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). BERT-based rankers have substantially improved the effectiveness of neural rankers.

Although effective to some extent, these rankers employ the cross-encoder architecture (as shown in Figure 1b) that is impractical to evaluate every passage or document 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 by 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 a dense retriever with the dual-encoder architecture.

## 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 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 in total, 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 span $p_i^{(s)}$, $p_i^{(s+1)}$, ..., $p_i^{(e)}$ from one passage $p_i$ that can answer the question, among the $M$ candidate passages. In this paper, we mainly focus on how to retrieve the passages that contains the answer via a dense retrieval approach.

### 3.2 The Dual-Encoder Architecture

We develop our approach based on the typical dual-encoder architecture. Figure 1a illustrates a dual-encoder. 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 that are the closest to the question in terms of the similarity in the embedding space will be retrieved. The similarity between the question $q$ and a candidate passage $p$ can be computed as the dot product of their vectors:

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

In practice, the separation of question encoding and passage encoding is desirable, so as the dense representations of all passages can be pre-computed for efficient retrieval. Here, we adopt two independent neural networks initialized from pre-trained LMs 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 following loss function:

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

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

**Inference** Since there is usually a large number of candidate passages at inference stage, a commonly adopted acceleration method is the exact maximum inner product search (Shrivastava and Li, 2014), and we use the implementation of FAISS (Johnson et al., 2019). The basic idea is to offline construct the index for the dense representations of all passages, and then at query time, we can find the top-$k$ most similar passages given the question embedding in sub-linear time.

### 3.3 Optimized Training Approach

In Section 1, we have discussed three major challenges in training dual-encoder based dense passage 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., 2019; Karpukhin et al., 2020; Luan et al., 2020). Assume that there are $B$ questions in a mini-batch on a single GPU, each of which 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. We present an illustrative

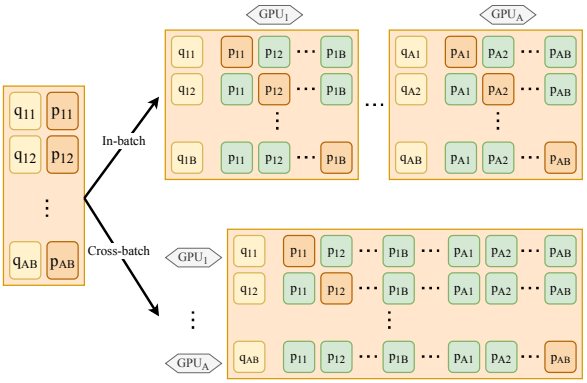



Figure 2: The comparison of traditional in-batch negatives and our cross-batch strategy when trained on multiple GPUs, where $A$ is the number of GPUs, and $B$ is the number of questions in each min-batch.

example at the top of Figure 2 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. Specially, 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 the examples (i.e., their dense representations) from other GPUs as the additional negatives for each question. Hence, with $A$ GPUs (or mini-batches), we can indeed obtain $A \times B - 1$ negatives for a given question, which is approximately $A$ times 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 optimization results are expected to be improved.

**Denoised Negative Sampling** Although the above strategy can increase the number of negatives, "easy negatives" are potentially incorporated, which can be easily discriminated by a passage retriever. While, hard negatives are shown to be more important to train a dual-encoder (Gillick et al., 2019; Wu et al., 2019; 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 positive passages) as negative samples (Karpukhin et al., 2020; Luan et al., 2020; Xiong et al., 2020). However, it is likely to contain false negatives, since the annotators can only annotate the a few top passages retrieved by a specific retriever. In our experiments on MSMARCO, we manually examine the top-ranked passages retrieved by our dense retriever, and we find that 70% of the passages that were not originally labeled as positives are actually positives or highly relevant. 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 does not 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.

Specifically, we first train a cross-encoder (following the architecture shown in Figure 1b) based on original training data. The learned cross-encoder can measure the similarity between questions and passages. Then, when sampling hard negatives from the top-ranked passages retrieved by a dense retriever, we remove the passages that are predicted as positives by the cross-encoder with high confidence scores. The cross-encoder architecture is more powerful for capturing two-way semantic interaction and shows much better performance than the dual-encoder architecture (Luan et al., 2020), while it is extremely inefficient over a large number of candidates in inference. Here, our idea is to utilize a well-trained cross-encoder to remove top-retrieved passages that are likely to be false negatives (i.e., unlabeled positives). The left 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. We first learn the cross-encoder with the original training data. 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.

### 3.4 The Training Procedure

As shown in Figure 3, we organize the above three optimization strategies into an effective training

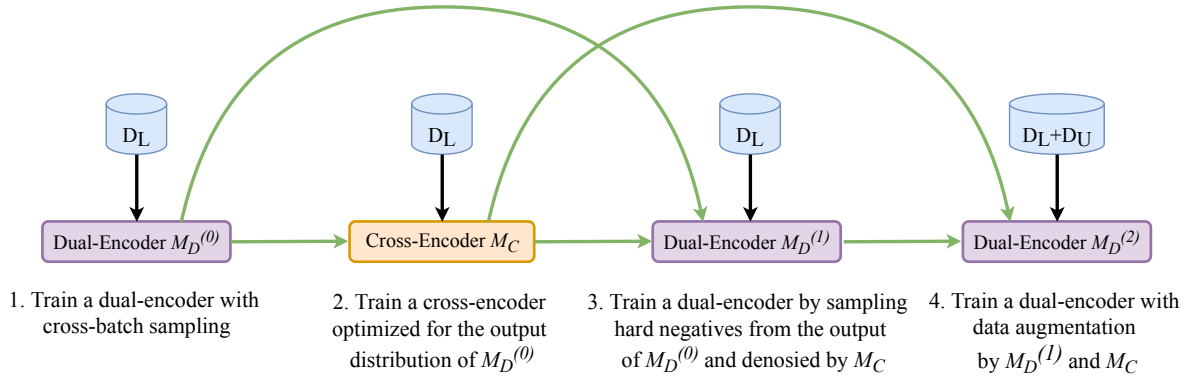



Figure 3: The pipeline of the optimized training approach RocketQA. Here, $M_D$ and $M_C$ denote the dual-encoder and cross-encoder, respectively. We use $M_D^{(0)}$, $M_D^{(1)}$ and $M_D^{(2)}$ to denote the learned models after different steps for the dual-encoder.

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 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 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 the passages that are predicted as positives by the cross-encoder $M_C$ with high confidence scores will be removed (i.e. denoised).
- **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-encoder. The cross-encoder is used in two steps with different purposes to promote the performance of the dual encoder, namely denoising hard negatives (STEP 3) and data augmentation (STEP 4).

## 4 Experiments

In this section, we first set up the experiments, and then present the results and analysis.

### 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 two datasets are constructed from search logs, reflecting the actual user needs in real life. The statistics of the datasets are listed in Table 1.

**MSMARCO Passage Ranking** MSMARCO is a large-scale dataset focused on open-domain QA, in which questions were sampled from Bing search logs. Based on the passages and questions in MSMARCO Question Answering, a dataset for passage ranking was created, namely MSMARCO Pas-

| 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.

sage 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 in (Karpukhin et al., 2020).

#### 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 finding.

**MRR** The Reciprocal Rank (RR) calculates the reciprocal of the rank at which the first relevant passage was retrieved. When averaged across questions, the measure 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 for which the top $k$ retrieved passages contain (at least) an answer.

**Exact match** This metric measures the percentage of predicted answers that match any one of the reference answers exactly, after string normalization such as article and punctuation removal.

#### 4.1.3 Implementation Details

We perform 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 ERNIE 2.0 base (Sun et al., 2019), 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.

**Batch size** The dual-encoders are trained with the batch sizes of $512 \times 8$ and $512 \times 2$ 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 $64 \times 4$ and $64$ on MSMARCO and NQ, respectively. Our implementation is based on FleetX (Dong, 2020) that is a highly scalable distributed training engine of PaddlePaddle. We use the automatic mixed precision and gradient checkpoint [3] functionality in FleetX, so as we can train the models using large batch sizes with limited resources. The cross-batch negative sampling is implemented with differentiable all-gather operation provided in FleetX. 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.

**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.

**Optimizer** We use LAMB optimizer (You et al., 2019) to train the dual-encoder on MSMARCO, since the batch size is large. In other settings, we always 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 $1e-5$.

**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 negatives used for training cross-encoders are randomly sampled from top-1000 passages retrieved by the dual-encoder $M_D^{(0)}$. 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. We use the cross-encoder for

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

| Methods | 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) | - | - | - | - | 78.4 | 85.4 |
| ANCE (single) (Xiong et al., 2020) | 33.0 | - | 95.9 | - | 81.9 | 87.5 |
| ME-BERT (Luan et al., 2020) | 33.8 | - | - | - | - | - |
| RocketQA | **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.

denoising hard negatives. Specifically, we select the top retrieved passages with a score higher than 0.9 as positive examples and those with a score less than 0.1 as negative examples.

**Unlabeled Questions** We collect 1.5 million unlabeled questions from Yahoo! Answers[4] and ORCAS (Craswell et al., 2020). We will make the questions publicly available with our code.

### 4.2 Experimental Results

In our experiments, we first examine the effectiveness of our retriever on MSMARCO and NQ datasets. Then, we conduct ablation study 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. Lastly, we report that RocketQA achieves the first place on the leaderboard of MSMARCO Passage Ranking.

#### 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 retrievers for comparison baselines.

The sparse retrievers includes the traditional retriever BM25, and four traditional retrievers enhanced by neural networks, including doc2query, DeepCT, docTTTTTquery and GAR. 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 includes DPR, ME-BERT and ANCE. 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 NQ datasets. Another important observation is that the dense retrievers are overall better than the sparse retrievers. Such a finding has also been reported in prior 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 Ablation Study

In this part, we conduct the ablation study on MSMARCO dataset to examine the effectiveness of the three strategies in RocketQA.

In Table 3, we organize the results by two main categories, i.e., whether hard negatives are used (the top part in Table 3) or not (the bottom in Table 3). Furthermore, we report the performance with respective to some specific strategy in each main category. We can make the following findings.

First, we compare cross-batch sampling with in-batch sampling 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 sampling. It is expected that when increasing the number of random negatives, it will reduce the discrepancy between training and inference and take fewer epochs and steps to reach a given accuracy. Furthermore, we investigate the effect of the number of random negatives, and examine the performance of dual-encoders trained by using different numbers of random negatives with a fixed step number. From Figure 4, we can see

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

| Strategy | MRR@10 |
|---|---|
| In-batch random negatives | 32.39 |
| Cross-batch random negatives | 33.32 |
| + Hard negatives w/o denoising | 26.03 |
| + Hard negatives w/ denoising | 36.38 |
| + Data aug (i.e. RocketQA) | **37.02** |

Table 3: The ablation study of RocketQA on MS-MARCO Passage Ranking.

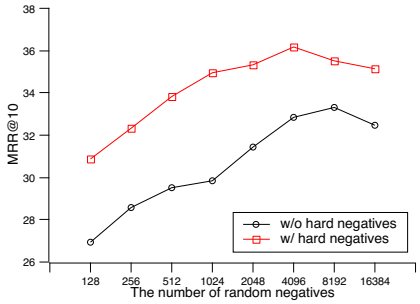



Figure 4: The effect of the number of random negatives paired for a question on MSMARCO dataset. The models without hard negatives are trained with 20K steps, and the models with hard negatives are trained with 5K steps.

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 the training data with limited size.

Second, we examine the effect of denoised hard negative sampling 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 hard negative sampling without denoising. We speculate that it is caused by the fact that there are a large number of unlabeled positives. For MSMARCO dataset, there is 1.1 labeled positives for each question on average. In order to check our speculation, we manually examine the top 100 questions passages returned by the retriever that were not labeled as positives in ground-truth data. We find that about 70% of them are actually positives or highly relevant. It is likely to bring noise if we simply sample from the top retrieved passages by the dense retriever, which is a widely adopted strategy to enhance the quality of negatives in prior studies (Gillick et al., 2019; Wu et al., 2019; Xiong et al., 2020). As a comparison, our proposed denoised hard negative sampling

| 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., 2019) | 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** |

Table 4: 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.

(the fourth row in Table 3) can help improve the performance of the dense retriever by reducing the influence of noise.

Finally, when integrated with the proposed 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 a more powerful capacity than the dual-encoder) to generate pseudo training data for improving the dual-encoder.

Results on NQ dataset has shown the similar findings on cross-batch sampling and denoised hard negative sampling (except for data augmentation) and omitted here due to limited space. We did not observe positive results for data augmentation on NQ dataset. A possible reason is that the unlabeled questions contain a large portion of non-factoid questions, while NQ dataset mainly focuses on factoid questions.

#### 4.2.3 Passage Reading with RocketQA

Previous experiments have shown the effectiveness of passage retrieval. Next, we further verify whether our RocketQA can improve the performance of passage reading for identifying correct answers.

We implement an end-to-end QA system in which we have a reader stacked on our RocketQA retriever. For a fair comparison, we re-use the released model [5] of the extractive reader in DPR (Karpukhin et al., 2020), and take 100 retrieved passages during inference (the same setting used in DPR).

Table 4 summarizes the the end-to-end QA performance of our approach and a number of com-

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

| Rank | Model | Date | MRR@10 on Eval | MRR@10 on Dev |
|---|---|---|---|---|
| 1 | RocketQA | Sep 18, 2020 | 42.6 | 43.9 |
| 2 | UED-large (Anonymous) | Aug 12, 2020 | 42.4 | 43.6 |
| 3 | DR-BERT (Sun et al., 2020) | May 20, 2020 | 41.9 | 42.0 |
| 4 | expando-mono-duo-T5 (Nogueira et al., 2020) | May 19, 2020 | 40.8 | 42.0 |
| 5 | DeepCT + TF-Ranking (Han et al., 2020) | Jun 2, 2020 | 40.7 | 42.1 |

Table 5: The leaderboard of MSMARCO Passage Ranking (by Oct 16, 2020), which can be visited at https://microsoft.github.io/msmarco/#ranking.

petitive methods. The performance is measured by the EM metric with the reference answer after minor normalization as in DPR. From Table 4, 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.

#### 4.2.4 Performance at the Shared Task of MSMARCO Passage Ranking

To further verify the effectiveness of our approach RocketQA, we participate the shared task of MSMARCO Passage Ranking held by Microsoft. Before our submission, there were already 103 submission at the leaderboard, where the top team achieved a performance record of 42.4 for MRR@10 on the evaluation set. To enhance the performance of our system, we also add a re-ranker to rerank the top retrieved results by the dual-encoder trained using RocketQA. Such a strategy has been commonly adopted in practical systems (Yan et al., 2019; Han et al., 2020). The re-ranker follows the cross-encoder architecture, and we ensemble two re-rankers initialized with ERNIE 2.0 large and Roberta large (Liu et al., 2019), respectively. We present the performance of our solution on Sep 18, 2020 (see Table 5), a new record of 42.6 for MRR@10, which takes the first place among the 104 submissions. The results have also demonstrated the effectiveness of our approach.

## 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 negative sampling and data augmentation. Extensive experiments have shown the effectiveness of the proposed approach by incorporating the three optimization strategies. Especially, built upon RocketQA, we have achieved the first rank at the leaderboard of MSMARCO Passage Ranking Task. In the future, we will work on training the whole QA system in a more end-to-end fashion.

## References


Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2019. Learning to retrieve reasoning paths over wikipedia graph for question answering. *arXiv preprint arXiv:1911.10470*.

Eric Brill, Susan 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)*, 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. *arXiv preprint arXiv:2002.03932*.

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 (Volume 1: Long Papers)*, pages 1870–1879.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. 2016. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*.

Nick Craswell, Daniel Campos, Bhaskar Mitra, Emine Yilmaz, and Bodo Billerbeck. 2020. Orcas: 18 million clicked query-document pairs for analyzing search. *arXiv preprint arXiv:2006.05324*.

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*, pages 985–988.

Hoa Trang Dang, Diane Kelly, and Jimmy J Lin. 2007. Overview of the trec 2007 question answering track. In *Trec*, volume 7, page 63.

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

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

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. 2010. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79.

Luyu Gao, Zhuyun Dai, and J. Callan. 2020. Modularized transfomer-based ranking framework.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. *arXiv preprint arXiv:1909.10506*.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 55–64.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Shuguang Han, Xuanhui Wang, Mike Bendersky, and Marc Najork. 2020. Learning-to-rank with bert in tf-ranking. *arXiv preprint arXiv:2004.08476*.

Matthew Henderson, Rami Al-Rfou, B. Strope, Yun-Hsuan Sung, L. Lukács, R. Guo, S. Kumar, B. Miklos, and R. Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *ArXiv*, abs/1705.00652.

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 *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2553–2561.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

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

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

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. *arXiv preprint arXiv:2004.12832*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

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

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

Jimmy Lin. 2019. The neural hype and comparisons against weak baselines. In *ACM SIGIR Forum*, volume 52, pages 40–51. ACM New York, NY, USA.

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

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2020. Sparse, dense, and attentional representations for text retrieval. *arXiv preprint arXiv: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. *arXiv preprint arXiv:2009.08553*.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019a. A discrete hard em approach for weakly supervised question answering. *arXiv preprint arXiv:1909.04849*.

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

Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1291–1299.

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. *arXiv preprint arXiv:1901.04085*.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713*.

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

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

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019c. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

A. Shrivastava and P. Li. 2014. Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips). *ArXiv*, abs/1405.5869.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2019. Ernie 2.0: A continual pre-training framework for language understanding. *arXiv preprint arXiv:1907.12412*.

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. *arXiv preprint arXiv:1908.08167*.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv:1911.03814*.

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*, pages 55–64.

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. *arXiv preprint arXiv: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 *TREC*.

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*, pages 1253–1256.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2019. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*.