Passage-based BM25 Hard Negatives: A Simple and Effective Negative Sampling Strategy For Dense Retrieval

Thanh-Do Nguyen [1], **Chi Minh Bui** [1], **Thi-Hai-Yen Vuong** [2] and **Xuan-Hieu Phan** [2]
[1] Viettel Cyberspace Center, Viettel Group, Hanoi, Vietnam
[2] VNU University of Engineering and Technology, Hanoi, Vietnam
{dont15, minhbc4}@viettel.com.vn, {yenvth, hieupx}@vnu.edu.vn

Abstract

Retrieval using dense representations has shown great capacity in capturing semantic similarities of the texts, but relies on high-quality selection of hard negatives for training (Karpukhin et al., 2020). This paper proposes a new hard negative mining strategy, called *passage-based BM25* (PassageBM25), to improve the performance of dense retrieval models. PassageBM25 is a new static hard negatives mining strategy that selects negative passages based on their similarity to the positive passage instead of the query. Empirical studies on ZAC2022, a Vietnamese question answering dataset, show that this approach is effective in such low-resource language and outperforms both the vanilla BM25 and dense retriever trained with conventional query-based BM25 method in terms of top-k retrieval accuracy. Furthermore, hard negatives mined with our proposed method can be used as a supplement to query-based BM25 hard negatives to enhance the retrieval performance, both in retriever only and retriever-reranker settings.

1 Introduction

Dense retrieval (Karpukhin et al., 2020; Qu et al., 2021) has emerged as a highly effective method for retrieving documents based on their semantic similarities. This method has been widely applied to various tasks such as web search, question answering, and fact verification. The reason for its success lies in its ability to capture the underlying meaning of text, which is often missed by traditional keyword-based search engines.

Researchers have proposed several methods to enhance the efficacy of dense retrieval models. These methods include distillation, retrieval-oriented pre-training, and negative mining. Distillation involves training a smaller model to replicate the behavior of a larger model, while retrieval-oriented pre-training entails training a language model on specialized tasks to enhance the vector space of text representations. Negative mining, on the other hand, involves selecting irrelevant documents, along with relevant ones, to provide the model with the most informative data for training.

One of the most significant challenges in dense retrieval models is the selection of high-quality hard negative documents. These documents have high similarity with the query but do not contain the answer to it. Random selection of these documents can lead to poor performance (Karpukhin et al., 2020). Meanwhile, one of the most popular *static hard negatives* is *query-based BM25* (QueryBM25) method (Zhao et al., 2022), which is fast to compute and gives reasonable performance. Although more advanced methods, often refered as *dynamic hard negatives* mining, can theoretically select optimal hard negatives, they require periodic index updates, which can be time-consuming and computationally expensive.

In the same line with the *static hard negatives* mining strategies like QueryBM25 approach, we propose the *passage-based BM25* (PassageBM25) method, that provides a simple and effective hard negative mining strategy that can improve the performance of dense retrieval models. Our method involves selecting negative passages based on their similarity to the positive passage, rather than the query. Empirical studies have demonstrated that this approach is effective in low-resource language settings such as Vietnamese and outperforms the conventional static method query-based BM25 without requiring expensive periodic index updates. Additionally, it has been shown that using hard negatives from our proposed strategy as a supplement to query-based BM25 hard negatives can further improve the performance of dense retrieval.

2 Related Work

Dense Retrieval The pretrained language models using Transformer architecture (Liu et al., 2019; Devlin et al., 2019) have shown the effectiveness in understanding natural language. Cross-encoder architecture (Nogueira and Cho, 2020) is an early adaption of these models for retrieval, which yields great results but computationally expensive. In contrast, dense retrieval uses a bi-encoder architecture which was first proposed by (Reimers and Gurevych, 2019) and then soon adopted for retrieval problems (Karpukhin et al., 2020; Lee et al., 2019). This method compares encoded query vectors with corpus document vectors using inner product. Dense retrieval pre-encodes the corpus into MIPS index to achieve search latency in milliseconds, using software like FAISS, Milvus.

Effective dense retriever Various techniques are used to improve the performance of dense retrievers. One line of works questions the capacity of single vector representation and proposed to use multi-vector representation (Khattab and Zaharia, 2020; Zhang et al., 2022). (Izacard and Grave, 2022) uses more sophisticated knowledge distillation technique, involving first training a teacher model and use its predictions at training time to optimize the dense retriever. Another approach attempts to pretrain models tailored for dense retrieval, either by adding an auxiliary training objective (Gao and Callan, 2021) or generating pseudo labeled data (Xu et al., 2022). In the same line of work as ours, many research has proposed methods for carefully selecting sets of negative samples used for training bi-encoder.

Hard negatives Hard negatives refer to the irrelevant texts but having a high semantic similarity with the query, which has been shown to improve the bi-encoder's capacity in discriminating between relevant and relevant texts. Following (Zhao et al., 2022), we categorize hard negatives into 2 types: static and dynamic. Static hard negatives are selected using a fixed negative selector during the whole training process. A straight forward static hard negative mining is to sample lexically similar texts (but does not contain the answer) returned by BM25, employed by (Karpukhin et al., 2020; Xiong et al., 2020). This strategy is often fast and only need to compute once. Later works explores dynamic hard negative mining techniques, such as using a learned retriever to mine hard negatives and re-train another retriever with them. The ANCE approach (Xiong et al., 2020) proposes to sample from the top retrieved texts by the optimized retriever itself, along with an asynchronous index refresh mechanism during training, which is very time-consuming. Another method ADORE (Zhan et al., 2021), fixes the text encoder and the text embedding index, and only utilizes an adaptive query encoder to retrieve top ranked texts as hard negatives. In general, dynamic negatives can provide more informative negatives during training at the cost of periodic indexing. Furthermore, RocketQA (Qu et al., 2021) proposes an optimal pipeline for training bi-encoder models, which involves increase batch size through in-batch and cross-batch training and denoise false negative samples.

Our PassageBM25 technique aims to provide a new static hard negative mining strategy that will not be computationally prohibitive. The effectiveness of PassageBM25 when used individually and when combined with query-based hard negatives will be shown in experiments.

3 Methodology

3.1 Preliminaries

Dense retrieval is a technique that involves encoding textual data into dense vectors with high dimensionality, and then measuring similarities between them using basic similarity functions. Pretrained language models, like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), are the key components used to encode texts into dense vectors that can capture their semantic meaning. During search time, the relevance scores of a query text with documents in the corpus can be computed using the dot product or cosine of their vector representations. Specifically, given a query q and a document d, two encoders, E_q and E_d, are utilized to map them to vectors v_q and v_d in R^d. The similarity between q and d is defined as follows:

$$sim(q, d) = f_{sim}(E_q(q), E_d(d)) \tag{1}$$

where f_{sim} is dot product or cosine function.

Training a good bi-encoder involves learning a good vector space such that representations of semantically similar queries and documents are clustered close to each other, whereas irrelevant queries and documents stay distant. The negative log likelihood of the positive passage is optimized as the loss function:

$$L(q, d^+) = -\log \frac{e^{\mathrm{sim}(q,d^+)}}{e^{\mathrm{sim}(q,d^+)} + \Sigma_{d^- \in NP} e^{\mathrm{sim}(q,d^-)}} \tag{2}$$

in which, q is a query, d^+ is its positive passage and $NP = \{d_1^-, d_2^-, \ldots, d_n^-\}$ are the negative passages.

A number of studies (Karpukhin et al., 2020; Xiong et al., 2020; Qu et al., 2021) have demonstrated the importance of selecting a high-quality set of negative passages when training an effective bi-encoder. Specifically, the inclusion of hard negatives in the NP set is often desired, as these are documents that exhibit high similarity with the query but are ultimately irrelevant.

Various strategies for mining hard negatives have been developed to improve dense retrieval performance. Among them, the most basic approach is to randomly select passages as hard negatives, but this method has been shown to be inferior (Karpukhin et al., 2020). Several studies have used a different approach, whereby they sample documents retrieved by BM25 that are similar in lexicon to the query, but do not contain the answers (Karpukhin et al., 2020; Xiong et al., 2020). For simplicity, we call this approach query-based BM25 or **QueryBM25** for short. A more sophisticated approach proposed by (Xiong et al., 2020) is to sample hard negatives from the top retrieved texts by the optimized retriever itself, which is theoretically more effective but requires periodic index updates that are computationally expensive. Each of these methods has its own tradeoffs in terms of effectiveness and computational cost that must be carefully considered.

3.2 Passage-based BM25

We provide an additional perspective on the source of hard negatives that should be considered when training a retrieval model. Our approach is a simple and effective strategy that can be used independently or as a supplement to existing hard negatives in training data.

Our approach is based on the hypothesis that the retrieval models can benefit from the ability to distinguish the relevant passage to a query among similar passages. Specifically, given a query q, we define p^+ and p^- as its positive and negative passage, respectively. If p^+ and p^- are semantically similar and their nuanced differences can only be discerned in the context of q, then the ranking ability of the model can be enhanced by learning to identify these differences.

In this study, we propose a new method called **PassageBM25**, which leverages the lexical overlap of passages, which are commonly and uniformly long. *Our hypothesis is that if these passages share a high degree of lexical overlap, they are likely to be semantically similar as well.* Based on this hypothesis, PassageBM25 involves sampling the best passages returned by BM25 (that do not contain the answers) using the positive passage as the query. In comparision, this approach differs from the conventional use of QueryBM25 in which the positive passage is used as the query. The detailed procedure is described in Algorithm 1.

```
Algorithm 1: PassageBM25
  Input: $query$, $positive\_psg$, $answer$,
         $retriever$, $topk$
  Output: $L = \{p_1, p_2, ..., p_k\}$ ($topk$ hard
          negative passages)

  /* the only difference from
     QueryBM25.                          */
  $text = positive\_psg$
  $C = retriever.retrieve(text)$

  /* filter out candidates containing
     the answer for the query            */
  $cands = [x$ for $x$ in $C$ if $answer$ not in $x]$
  $result = cands[: topk]$

  return $result$
```

Algorithm 1 requires a positive passage ($positive\_psg$) of a query as input. The short answer ($answer$) for the query is utilized to exclude passages that may have the answer. However, in scenarios where no short answer is annotated, the entire $positive\_psg$ can be used as the answer for the exclusion filter.

4 Experimental Setup

In this section, we describe the data we used for experiments and the basic setup.

4.1 Wikipedia Data Pre-processing

We use the Vietnamese Wikipedia dump from June. 20, 2022 as the source documents for answering questions. The Vietnamese Wikipedia dump was obtained from the Zalo AI Challenge 2022 competition and was provided in two formats: a raw dump and a cleaned version that only included text portions.

We partition each Wiki article into multiple, separate text blocks with the following procedure. First, we use the `sent_tokenize` function provided by the `underthesea` python package to split a wiki document into sentences. These sentences are then grouped into passages, with each passage containing no more than 100 words. If a sentence exceeded this limit, we truncate it accordingly. Ultimately, we obtain approximately 3,000,000 passages, which serve as our basic retrieval units.

4.2 Retrieval Dataset

Zalo End2end Question Answering (ZAC2022) dataset (zal) was specifically designed for the purpose of building open domain question answering systems in the ZAC competition. Each of sample in the dataset includes a question and corresponding context passages that were annotated from the Vietnamese Wikipedia Corpus, making up 7114 questions and 20857 context passages in total.

Since the original context passages have been generated differently than passages in our processed passage corpus, we match and replace each gold passage with the corresponding passage in our corpus. With an input question, we use BM25 to find a list of the most lexically similar passages with its gold passage, then select the first one in this list which contains the short answer to the input question. If no short answer is provided for that question, we simply select the first one out of the list. The resulting processed dataset serves as the only source of data for training retreival model.

Additionally, the bi-encoder model requires each data sample to include a positive context passage to calculate the negative log likelihood loss during training. However this property is not ensured for the ZAC2022 dataset, as each sample in this dataset can be categorized into one of three different types. These include: Type 1 - contains at least one positive context along with an annotated short answer; Type 2 - contains at least one positive context, but there is no annotated short answer; and Type 3 - contains only negative contexts that do not provide an answer to the question. Therefore, we can only make use of Type 1 and Type 2 samples for training. When conducting k-fold cross-validation (k=5) to achieve a fair evaluation, we make sure the number of samples belonging to types (1), (2), and (3) is equally distributed to each fold, as shown in Table 1.

Fold	0	1	2	3	4
#Type 1	771	771	770	770	770
#Type 2	517	517	517	517	517
#Type 3	136	136	135	135	135
Total	1424	1424	1422	1422	1422

Table 1: Number of sample types in 5-folds

4.3 Machine Reading Comprehension Dadasets

We are interested in the impact of PassageBM25 on the end-to-end question-answering system, therefore we train an additional reading comprehension model to complete the pipeline. This model takes the question and retrieved passages as inputs, and then extracts spans as the the answers. In the following, we provide details on datasets used for training that baseline extractive reader. For clarification, we do not use training/test splits and corpus from the following datasets (except for **ZAC2022**) for training/testing retrieval models.

In addition to the **ZAC2022** dataset, which comprises 8487 passages paired with answerable questions and 12370 passages paired with unanswerable questions, we gathered as many Vietnamese question-answering datasets as possible. These datasets include:

MLQA (Lewis et al., 2020) was released by Facebook (Meta) and contains over 5,000 question-answer pairs in the SQuAD format across seven different languages, including Vietnamese.

XQUAD consists of 1,190 question-answer pairs from the SQuAD v1.1 development set (Rajpurkar et al., 2016), along with their professional translations into ten languages, including Vietnamese.

UIT ViQuAD 2.0 includes 23,000 question-answer pairs from UIT-ViQuAD 1.0 (Nguyen et al., 2020) and over 12,000 unanswerable questions.

We then obtain a joint dataset comprising over 30,000 text passages and nearly 60,000 questions (both answerable and unanswerable). Number of questions and context passages in training and testing splits are shown in table 2

	#Questions	**#Contexts**
Train	56795	28959
Dev	4208	2493

Table 2: Number of questions and contexts in training and development splits.

4.4 Compared Systems

In training time, to thoroughly evaluate the effectiveness of our proposed negative sampling strategy, we experimented bi-encoders and cross-encoders with three training strategys. The first strategy is the conventional QueryBM25, while the second strategy utilized our proposed PassageBM25, enabling us to assess the effectiveness of our method independently. The third strategy combines two previous approaches: half of the negative samples were selected from the first strategy, while the other half were selected from the second strategy. This allowed us to assess the benefits of using our method as a supplement to existing training data.

During inference phase, we test various configurations of a retrieval pipeline consisting of a first-stage retriever and a subsequent reranker to identify the optimal setup. Each configuration is different in terms of the type of its retriever and reranker, and the specific details for each configuration can be found in Table 3.

Table 3: Retrieval pipeline configurations

Retriever	Reranker
BM25	-
Bi-encoder	-
BM25	Bi-encoder
BM25	Cross-encoder

4.5 Implementation

The Elasticsearch[1] software is utilized to implement the BM25 retrieval method in our experiments. We initialize all bi-encoder and cross-encoder models from the vinai/phobert-base checkpoint, which is a robust language model for Vietnamese (Nguyen and Tuan Nguyen, 2020). For training all bi-encoders, we use the DPR implementation of the haystack library, enable in-batch negatives, set the number of hard negatives for each question to 8, with a batch size of 8. We train for 5 epochs, and for the rest of the hyperparameters, we adopt the settings used in (Karpukhin et al., 2020). For training cross-encoders, we use the transformers library and treat the text ranking problem as a regression problem with a score of 1.0 for pairs of questions and relevant passages and a score of 0.0 for the inverse case. For each question, we sample 1 positive passage and 29 negative passages, with a batch size of 16, and train each cross-encoder for 10 epochs.

For training the reader model, we use xlm-roberta checkpoint to train for 2 epochs, with batch size of 32, learning rate 2e-5 and the rest of the hyperparameters follow (Zhang et al., 2020). All our experiments were conducted on a single 32GB V100 GPU.

4.6 Evaluation Metrics

Accuracy@k is a metric used to evaluate the retrieval accuracy with a fixed value of k. Given the the retrieved collection of passages C, it measures the percentage of questions q (out of all questions) whose C contains at least one passage containing the answer to q. For a test set of questions Q, the retrieval accuracy top-k is calculated as follows:

$$Accuracy@K(Q) = \frac{1}{|Q|} \sum_{i=1}^{|Q|} hit(q_i, C_{q_i})$$

where q_i is the ith question in Q, C_{q_i} is the set of K corresponding candidate passages. $hit(q_i, C_{q_i})$ can take the value of 0 or 1, calculated specifically as follows:

$$hit(q_i, C_{q_i}) = \mathbf{I}\{(P^+ \cap C) \neq \emptyset\}$$

where P^+ is the set of passages that can answer q.

To evaluate the performance of a reader model, we use the EM and F1 scores. EM measures the exact match between model and ground truth, while F1 score measures the balance between precision and recall.

5 Retrieval Results

In this section, we assess the retrieval performance of our PassageBM25 negative sampling strategy on various retrieval settings. All the results presented in this section are the average of results from cross-validation with 5 folds. We also conduct an analysis to determine how its result differs from the QueryBM25 approach. Additionally, we provide information on the runtime efficiency of each retrieval setting.

Retriever only pipeline. Table 4 presents the results of the retrieval phase using only retriever. The retrieval accuracy using only a bi-encoder is not impressive as expected. Unlike the reported accuracy@20 results of 9% - 19% better than BM25 in (Karpukhin et al., 2020), we trained a bi-encoder with the negative passages

[1] https://www.elastic.co/

Retrieval phase		Accuracy@K						
Retriever	**Reranker**	1	3	5	10	20	50	100
BM25	-	24,40	39,01	45,88	54,91	63,15	72,48	78,27
Bi.+strategy_1	-	20,63	34,30	40,81	49,43	56,92	66,95	72,90
Bi.+strategy_2	-	24,67	40,07	46,73	56,11	63,80	73,15	**79,10**
Bi.+combined	-	**25,58**	**41,26**	**48,71**	**58,00**	**65,76**	**73,83**	79,05

Table 4: Accuracy@k of retrieval phase with retriever-only pipeline

Retrieval phase		Accuracy@K						
Retriever	**Reranker**	1	3	5	10	20	50	100
BM25	Bi.+strategy_1	28,53	47,32	55,43	64,24	71,41	77,26	**78,27**
	Bi.+strategy_2	29,74	48,19	56,11	65,45	72,13	77,04	**78,27**
	Bi.+combined	**32,30**	**51,65**	**59,37**	**67,25**	**73,32**	**77,26**	**78,27**
BM25	Cross.+strategy_1	36,27	53,35	59,09	65,15	70,76	76,12	**78,27**
	Cross.+strategy_2	35,29	**55,38**	**61,89**	**68,22**	**72,84**	**76,66**	**78,27**
	Cross.+combined	**37,83**	54,61	59,75	65,88	70,32	75,53	**78,27**

Table 5: Accuracy@k of retrieval phase with retriever-reranker pipeline. Bi. and Cross. denote a bi-encoder and cross-encoder model, respectively. The hard negative sampling strategies are denoted by strategy_1: QueryBM25 approach, strategy_2: PassageBM25 apprach, combined: combination of QueryBM25 and PassageBM25.

sampled from QueryBM25 (strategy 1), the results for Accuracy@{1-200} are lower than BM25 by 4% - 10%. We conjecture this difference comes from the fact that DPR train its model with significantly larger batch size (128) compared to ours (8), plus that we do not use the same starting checkpoint for training.

However, contrary to the obstacle of limited data, our bi-encoder model trained with data sampled from PassageBM25 (strategy 2) consistently gives results better than query-based BM25 by about 1% on Accuracy@{1-100}. The performance of the bi-encoder model continues to improve when using data sampled from combining query-based and passage-based BM25, with an average increase of 1% in accuracy for k $\in$ {1, 3, 5, 10, 20, 50}.

This suggests that our method enables trained bi-encoder to perform consistently better than QueryBM25 under limited Vietnamese training samples and computation.

Retriever-reranker pipeline. To capture the strength of both lexical and semantic-based approaches, we utilized the BM25 retrieval model as the first-stage retriever, followed by different rerankers (biencoder or crossencoder), with the results presented in table 5. Our experiments indicate that reranking the top 100 passages returned from BM25 yields the best results in most cases, therefore, we will only include these results in our reports.

When using the biencoder reranker trained with QueryBM25 approach, the Accuracy@k improved significantly by 4-11% compared to using the biencoder as the retriever only. PassageBM25 witnesses a consistently higher improvement of about 1% than with QueryBM25. Finally, the best results are given when using negative samples combined from both query-based and passage-based BM25. However, when using cross-encoders, model trained with PassageBM25 hard negatives achieves the best result. Cross-encoder, despite having richer interaction between query and passage, is outperformed by bi-encoders with Accuracy@{20, 50, 100}.

These results demonstrate that combining BM25 and biencoder leads to superior results compared to using either pure BM25 or biencoder. They also show the effectiveness of our proposed PassageBm25 approach when used individually and as a supplement to the conventional QueryBM25.

Runtime Effiency Table 6 demonstrates inference's runtime for each query in different retrieval phase settings. In terms of time, BM25 is the fastest retrieval method with an average running time of only 0.81 seconds. On the other hand, retrieving documents indexed with `faiss's IndexFlatIP` takes much more time, averaging 2.561 seconds.

However, when combining BM25 and biencoder, the performance is superior and the retrieval speed is only about 0.02 seconds slower than BM25. Finally, the re-ranking algorithm using crossencoders takes more than 4 times longer than BM25.

Retrieval phase		**Latency**
Retriever	**Reranker**	**(Unit s)**
BM25	-	0.081 ± 0.003
Bi-encoder	-	2.561 ± 0.154
BM25	Bi-encoder	0.104 ± 0.003
BM25	Cross-encoder	0.374 ± 0.003

Table 6: Runtime effiency

6 End To End Question Answering Results On ZAC2022

Our end to end question answering system consists of 2 main phases: retrieval phase and machine reading comprehension phase. The retrieved documents from the first phase (20 passages in our experiments) will be fed into a reader model in the second phase, with each passage this model will extract list of candidate spans that may answer to the initital question. The reader we used is an extractive reader, trained with the method of Retro-reader proposed in (Zhang et al., 2020). The EM and F1 scores of this model are shown in the 7. We then fix this model as our only reader in all experiments and plug different retriever and reranker for the retrieval phase.

To evaluate our end-to-end model, we used the public test set of Zalo AI Challenge which includes 600 questions. However, the evaluation process is different from usual, as we will submit our results through the competition's submission portal and we do not have access to the specific answers. The evaluation will be conducted on the competition's server-side. Table 8 shows the submission results of our experiments.

Overall, it can be seen that the end-to-end accuracy is proportional to the top 20 retrieval accuracy. With the retrieval phase consisting of only BM25, the accuracy is actually quite competitive, reaching 69.83%, while none of the single bi-encoder models surpass this score. When using a reranker, the accuracy is significantly improved and reaches the highest score of 76.17% with the bi-encoder trained with negative data sampled from strategy (3).

It is worth to note that the best reported top 20 retrieval accuracy in section 5.3 is 73.32%, while the accuracy of the end-to-end model evaluated on the public test set is 76.167%. This is unusual because the reading comprehension phase may not always extract the exact answer, thus the end-to-end accuracy is usually lower rather than the retrieval accuracy. However, it can be simply explained that these results are not evaluated on the same set of questions. Additionally, when evaluating the retrieval phase, we only consider one passage as relevant. In reality, there may be multiple passages containing the answer in the corpus, so our evaluation metric is more strict than reality.

7 Conclusion

In this work, we demonstrated that passage-based BM25 has the ability to outperform query-based BM25. Additionally, it can also supplement query-based BM25 hard negatives in order to improve the efficiency of training dense retrieval. As a result of improved dense retrievers, we obtained the best performance on the Vietnamese open-domain question answering dataset ZAC2022, in terms of both retrieval accuracy and end-to-end accuracy.

References

Zalo ai challenge - question answering track. `https://challenge.zalo.ai/portal/e2e-question-answering`. Accessed: 2023-07-30.

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

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

Gautier Izacard and Edouard Grave. 2022. Distilling knowledge from reader to retriever for question answering.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the*

Question type	**All questions**			**Answerable questions**			**Unanswerable questions**		
Score	EM	F1	Count	EM	F1	Count	EM	F1	Count
	72.48	83.76	4208	65.34	80.13	3208	95.4	95.4	1000

Table 7: Performance (EM and F1 scores) of reader model on our custom test set.

Retriever	**Reranker**	**Accuracy**	**Runtime (s)**
BM25	-	69.833	0.240 ± 0.003
Bi.+strategy_1	-	65.833	2.588 ± 0.042
Bi.+strategy_2	-	68.333	
Bi.+combined	-	69.5	
BM25	Bi.+strategy_1	74.333	0.265 ± 0.003
	Bi.+strategy_2	73.5	
	Bi.+combined	**76.167**	
	Cross.+strategy_1	73.333	0.539 ± 0.003
	Cross.+strategy_2	73.833	
	Cross.+combined	74.167	

Table 8: Accuracy of end-to-end question answering models on public test set of ZAC2022 competition.

2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 39–48, New York, NY, USA. Association for Computing Machinery.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

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.

Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. PhoBERT: Pre-trained language models for Vietnamese. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1037–1042, Online. Association for Computational Linguistics.

Kiet Van Nguyen, Duc-Vu Nguyen, Anh Gia-Tuan Nguyen, and Ngan Luu-Thuy Nguyen. 2020. A vietnamese dataset for evaluating machine reading comprehension.

Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage re-ranking with bert.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

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.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2022. LaPraDoR: Unsupervised pretrained dense retriever for zero-shot text retrieval. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3557–3569, Dublin, Ireland. Association for Computational Linguistics.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1503–1512, New York, NY, USA. Association for Computing Machinery.

Shunyu Zhang, Yaobo Liang, Ming Gong, Daxin Jiang, and Nan Duan. 2022. Multi-view document representation learning for open-domain dense retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5990–6000, Dublin, Ireland. Association for Computational Linguistics.

Zhuosheng Zhang, Junjie Yang, and Hai Zhao. 2020. Retrospective reader for machine reading comprehension.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2022. Dense text retrieval based on pretrained language models: A survey.