Building an Efficient Retriever System with Limited Resources

Quang Nhat Nguyen[1] and Huong Thanh Le[1]

[1] Hanoi University of Science and Technology, Hanoi, Vietnam
nhatquang2306@gmail.com, huonglt@soict.hust.edu.vn

Abstract. Despite significant advancements in Question-Answering (QA) systems based on Large Language Models (LLMs), the issue of generating imprecise answers leading to less informative responses persists. To develop effective QA systems for open-domain datasets, particularly content-specific datasets, dense passage retrieval and the two-stage retriever-reader model remain a rational choice. However, when being applied in real-world systems, these approaches encounter challenges posed by the limitation of computational resources and training data. To address the scarcity of training data, we propose fine-tuning the pretrained BERT-based encoder using masked language modeling before employing a dual-encoder architecture—an established and efficient technique. Additionally, we introduce a modified loss function for dual-encoder training that reduces memory usage during training without compromising system performance. The new loss function is employed in a multi-stage training strategy, yielding enhanced retriever performance at each training stage. To further augment the system's capabilities, we train a cross-encoder to construct a robust retriever for domain-specific datasets. The effectiveness of these proposed techniques is validated by experiments with significant increases in performance compared to the baseline models, underscoring their potential to advance the state-of-the-art in open-domain question-answering systems.

Keywords: Question answering, Information retrieval, Dual-encoder, Vietnamese legal texts.

1 Introduction

Nowadays, with the development of large language models (LLMs), question-answering systems can use pre-trained knowledge to generate answers to input questions without the need for a retrieval step. However, due to the vast amount of training data, these systems might provide inaccurate results when answering questions in a domain-specific area. Therefore, when building domain-oriented question-answering applications, Information Retrieval (IR) remains a critical step that determines the accuracy of the system.

There are two primary approaches to constructing a retrieval model: word matching and semantic searching. Semantic retriever is more optimal because of its ability to truly understand the question and documents to deal with acronyms and paraphrases that remain as the weakness of the word matching strategy. A typical semantic retriever uses a dual-encoder, which has been widely explored [8, 10, 13, 14, 18], to retrieve the most relevant documents for a query and then a cross-encoder [3, 10, 12, 13] to rank them further.

However, training an effective dual-encoder usually requires a large and diverse dataset, while real-world question-answering systems often focus on specific domains with limited data. This can lead to overfitting, where the model performs poorly because it's too specialized. Moreover, training a dual-encoder requires significant computing resources, which may not be readily available, making resource-intensive approaches impractical. Using limited computing resources results in smaller training batch sizes, reducing model performance. Additionally, the inappropriate percentage of hard negative passages in a batch can also affect the model's performance negatively.

This paper addresses these challenges faced by real-world question-answering systems. To handle limited data, we fine-tune the pre-trained encoder on domain-specific text using masked language modeling (MLM) [2]. We employ a multi-stage training strategy for the dual-encoder to enhance its accuracy and integrate a cross-encoder re-ranker to optimize the retriever's performance. To effectively navigate the limitations of parallel computing memory, we introduce a modified loss function in fine-tuning pre-trained encoders for dense passage retrieval, in order to reduce the impact of hard negatives, allowing the model to converge better. We identify and address the shortcomings of the existing dataset by expanding it through targeted web crawling and appending corresponding titles to the beginnings of all passages. Through comprehensive experimentation with these new passages, we achieve notable performance improvements. These enhancements highlight the importance of titles and reinforce the effectiveness of their innovative methods, offering valuable insights for real-world question-answering systems.

2 Background and Related Works

Since this paper concentrates on improving the performance of the retriever in the situation of lacking training data and computing resources, this section discusses researches involving the dual-encoder and the cross-encoder.

2.1 The dual-encoder passages retriever

The semantic search uses pre-trained language models to acquire semantic representations of queries and passages. Robust pre-trained models with extensive knowledge and learnability for every domain can efficiently encode questions and contexts with dense embedding vectors and assess their relevancy using similarity functions like cosine or inner product. The actual use of this strategy necessitates the saving of passage vectors as well as the use of an efficient index to retrieve relevant contexts rapidly. The dual-encoder [7], which utilizes two distinct deep encoders for questions and passages, is the most commonly investigated architecture for the semantic-encoding technique and provides the most stunning results. However, it often requires relatively large amounts of training data.

The dual-encoder's training processes include certain notable characteristics:

- There are two main approaches: self-supervised pre-training [1, 4, 5, 8] and fine-tuning on labeled question-passage pairs dataset. We employ the second approach, which saves parallel training resources.
- Hard negative passages (hard negatives) are contexts that do not contain the answers but have some levels of semantic or lexical relevance to the question. Utilizing an appropriate quantity of hard negatives in training helps the model better recognize the incorrect results, thus enhancing the model's performance [7, 10, 13, 18, 19].
- Updating hard negatives during training (dynamic hard negatives) [19] or between training stages (multi-stage training) [13] has been demonstrated to be an optimal technique. As word-occurrence vectors utilizing TF-IDF [17], BM25 [16] and variants are frequently used to choose hard negatives, updating hard negatives using the recently-trained model between the training steps is a powerful design that helps the model address their shortcoming without costing too much resources.

2.2 The cross-encoder passages re-ranker

Besides dual-encoder, cross-encoder is also shown as an outstanding method for information retrieval and question-answering problems [3, 10, 12, 13]. Unlike dual-encoder, cross-encoder only needs to use one pre-trained BERT-based encoder model in both training and inference procedures. The questions will be attached to paragraphs, separated by a special character and used as input to the model. Then the vector representation of the first token (e.g. [CLS] symbol in BERT) will continue to be put into a classifier to decide if this is a question-paragraph pair containing the answer or not. This is a classic method associated with the advent of pre-trained Transformer-based encoders but so far, it still gives good results, even better than dual-encoder [10]. However, using only a cross-encoder is an expensive and infeasible method in practice. Dual-encoder model is always preferred in information retrieval problems because of its high retrieving speed despite poorer accuracy. Cross-encoder architecture is currently applied in building a re-ranker - a model after the dual-encoder retriever to re-evaluate the top passages returned by the dual-encoder. We also built a cross-encoder re-ranker in our model to improve the performance of the whole retrieval system on the dataset.

3 Our Proposed Approach

In the context of a Question Answering (QA) system, the optimal objective is to provide precise answers for given queries. In this pursuit, it is better for the retriever to return passages instead of documents. To this end, our retriever utilizes a collection of passages, rather than entire documents, within the corpus.

The retrieval problem is formulated as follows: Given a question q and a corpus of passages C, the system needs to retrieve the passages that may contain the answer to the input question. Our proposed system is applied to the Vietnamese legal dataset.

Our retrieval system architecture is shown in **Fig. 1**, which includes two main components: the dual-encoder and the cross-encoder. The dual-encoder contains two separate pre-trained encoders: one for the question and another for the passage. PhoBERT [11], a pre-trained encoder for Vietnamese language is applied to encode the input text. PhoBERT was trained based on RoBERTa [9] which optimized the BERT pre-training procedure for more robust performance. The encoded passages are stored and indexed using the FAISS indexer [6] — a proficient algorithm for approximate similarity search, renowned for its scalability to billions of vectors. The similarity of the question and context is calculated as the dot product of their pair of embedding vectors. The high dot product score means that this is more likely a pair of a question and a passage containing the answer you are looking for. In this paper, we retrieve top k passages with k equal to 30. This ensures both the retrieval speed and the accuracy of the model. These passages are fed into the cross-encoder.

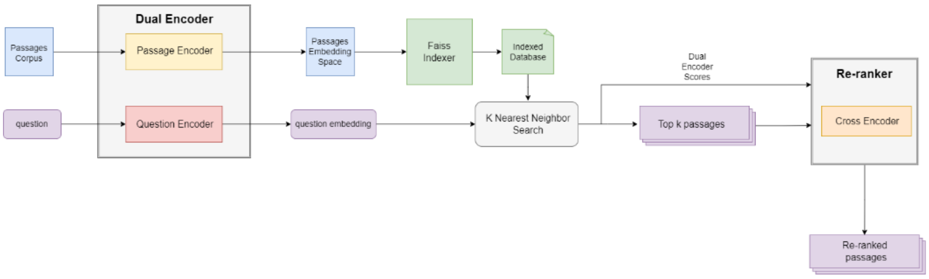

Fig. 1. Our retrieval system.

The cross-encoder includes the PhoBERT pre-trained encoder and a classifier which consists of a dropout, a fully connected and a softmax layer. The model will be trained to return 0(YES) if the passage attached to the question contains the answer, and 1(NO) otherwise.

The scores of the dual-encoder and cross-encoder are combined to form a final re-rank score for a passage:

$$score_{rerank} = \frac{score_{dual}}{100} + score_{cross} \tag{1}$$

After the k passages are re-ranked, we can choose a number of passages or set a score threshold to return the best paragraphs. The results of the retriever can be passed directly to the user (in case k is small) or further fed into other models (reader, generator...) to return the most concise answer to the question. The rest of this section will introduce our proposed strategies to improve the retriever's performance.

3.1 Fine-tuning PhoBERT for Domain-based QA systems

PhoBERT was pre-trained with 20GB of Wikipedia and News texts. It can well understand daily-life Vietnamese text. However, when working with data in specific domains such as law or bioinformatics, its performance is getting worse since each specific domain has a different vocabulary. Given a small closed-domain QA dataset, it is not good for just fine-tuning the PhoBERT with this dataset, since it is not enough for the pre-trained encoders to understand the domain problem. To solve this problem, we fine-tune the PhoBERT with the task masked-language modeling (MLM) on content passages of the dataset before training the dual-encoder. This training process helps the model learn closed-domain information for better semantic understanding in the encoding step. The fine-tuned PhoBERT is used to encode the input text of the QA system.

3.2 The Dual-encoder

Improve the loss function. Let us consider a training dataset $D = \{\langle q_i, p_i^+, p_{i,1}^-, \dots, p_{i,n}^- \rangle\}_{i=1}^{m}$ comprising m instances, each instance contains a unique question, q_i, and a pertinent (positive) passage, p_i^+, alongside n inconsequential (negative) passages, $p_{i,j}^-$. The original loss function given by Karpukhin et al. [7] is:

$$L_1 = -log \frac{e^{sim(q_i,p_i^+)}}{e^{sim(q_i,p_i^+)} + \sum_{j=1}^{n} e^{sim(q_i,p_{i,j}^-)}} \tag{2}$$

When using some in-batch hard negatives (i.e., $p_{i,k}^*$ for each instance), the loss function is now calculated as:

$$L_2 = -log \frac{e^{sim(q_i,p_i^+)}}{e^{sim(q_i,p_i^+)} + \sum_{k=1}^{m} e^{sim(q_i,p_{i,k}^*)} + \sum_{j=1}^{n} e^{sim(q_i,p_{i,j}^-)}} \tag{3}$$

According to Karpukhin et al. [7] and some other studies [10, 13, 18], utilizing at least one hard negative per instance increases the model's ability to distinguish between positive and negative passages, resulting in better retrieval results. Nevertheless, when dealing with the Vietnamese legal dataset, it has come to our attention that the model's performance deteriorates when one or a few hard negatives generated by BM25+ [15] are added. Upon an exhaustive investigation into the potential factors contributing to this regression, we have identified a potential source of this issue—discrepancies in batch size between our implementation and the original paper [7]. Karpukhin et al. [7] employed large batch sizes, such as 128 or more, facilitated by modern computing resources. Conversely, due to constraints related to parallel resources, our batch size configurations are limited at 64 and 32, particularly when incorporating an additional hard negative. This discrepancy leads to an imbalance in the ratio of hard negative passages to the overall count of negative passages within our implementation. Consequently, this skewed ratio results in overfitting of the model, causing it to excessively prioritize hard negatives at the expense of generalization capabilities.

So, how to still use hard negatives and achieve better results when working with limited parallel resources? We propose a newly modified loss function to control the influence of hard negative segments during training:

$$L' = -\alpha . L_1 + -(1 - \alpha). L_2 \tag{4}$$

The new loss function will comprise a weighted combination of two negative log-likelihood loss components: one incorporating hard negatives and the other exclusively accounting for random negatives. To balance these two loss functions, we propose an α hyper-parameter between 0 and 1. A good α value will sufficiently control the influence of hard negatives in the training process. $\alpha = 0$ means we are not using hard negatives; $\alpha = 1$ means we are actually just using a loss function that contains hard negatives. Different values of α are tested in our experiments in order to pick the optimal one. This will be presented in the experimental results section.

Choosing Efficient Training Samples. In addition to positive and negative samples, the integration of hard negatives proves instrumental in augmenting system performance. Methods such as TF-IDF [17], BM25 [16], and their variants are often used to select hard negatives for training. During the training process, to a certain point, the model was able to learn those hard negatives well but at the same time, could not distinguish some negatives with complex semantics. Therefore, it is very important to update the hard negatives. The new hard negatives will be the negatives that are mistaken by the current model as positives. Because of resource consumption, the update can only happen when we stop the training procedure.

3.3 Cross-encoder

The top k results of the dual-encoder are reranked by the cross-encoder, which works as a classifier. Positive samples of the classifier's training dataset are the passages corresponding to the question from the dataset. Negative samples are taken from the top negative passages returned by the dual-encoder. Passages that are longer than the maximum input length that PhoBERT can accept will be truncated to a reasonable length. Since one positive sample accompanies with several negative samples, we repeat the positive samples many times to balance the training data.

4 Experiments

4.1 Dataset

We utilize the Vietnamese legal dataset in the text retrieval task from the Zalo AI Competition 2021[1] for evaluating the system's performance in text retrieval. This dataset consists of a collection of legal documents organized into passages, accompanied by a training dataset containing individual questions along with

[1] https://challenge.zalo.ai/portal/legal-text-retrieval

corresponding answers, some of which may have multiple responses. Since only the training dataset is public by the Zalo AI Competition, we randomly divide 3196 samples in that set into 3 smaller train-val-test sets with the number of samples of 2400, 350, and 446 respectively. For the corpus of law passages, after filtering out duplicated ones, we get 60830 law passages from 3263 law documents.

4.2 Experimental Setup

Since the PhoBERT encoders are used in both the dual-encoder and cross-encoder, preprocessed questions and passages need to go through the word segmentation process before being fed into the models. $Pyvi$[2] - a word segmenter - is used for this task.

To get the hard negative passages for each question in the training process, we selected passages returned by BM25+ in $rank - bm25$[3] library, which are not the correct passages for each input question.

Dual-encoder two-stage training. After getting the initial hard negatives with BM25+, we train the dual-encoder in the first stage on 80 epochs. The last model is then saved and used to update hard negatives, which are used in the next 10-epoch-training stage. In the training procedure of dual-encoder, we set the learning rate for both stages as $1e^{-5}$ and use the Adam optimization with linear scheduling. We experiment with batch sizes from 16 to 64.

Cross-encoder training. We truncate long passages into smaller ones and let the trained dual-encoder prepare the training dataset for the cross-encoder as described before. We also use Adam optimization with linear scheduling and a dropout rate of 0.1. We set the batch size as 16 and train the model for 10 epochs, evaluating the model on the validation set after each 1000 training steps. The checkpoint of the model having the highest results on the validation set is saved. We restate the importance of data balancing through duplicating positive samples. Both the training procedures of cross-encoder and dual-encoder are conducted on Kaggle[4] with GPU P100 or two GPUs T4.

Evaluation metric. One feature of the Zalo legal question-answering dataset is that out of 3196 samples, there are 3103 with only one positive passage (accounting for more than 97%). Questions with two or three positive passages make up the remaining 3%, a rather modest number. Therefore, instead of using the popular recall score like other information retriever studies, we use the accuracy to measure the proportion of samples with at least one answering passage returned in the top k passages scored by the models. The formula for the measure is as follows:

[2] https://pypi.org/project/pyvi/

[3] https://pypi.org/project/rank-bm25/

[4] https://www.kaggle.com/

$$\boldsymbol{acc}_k = \frac{n_{p,k}}{n}.100\% \qquad (5)$$

where n is the number of samples in the evaluating dataset, $n_{p,k}$ is the number of samples having at least one positive passage retrieved in the top-k articles. We evaluate the dual-encoder with the top 1, 5, 10, 30, and 100 retrieved passages. With the cross-encoder re-ranker, we only consider the improvement in performance in the top 1, 5, and 10 after re-ranking the top 30 articles returned by the dual-encoder. We also recorded the results when the trained models inferred with the texts that were truncated to match the maximum input length of PhoBERT.

4.3 Experimental Results

In this section, we denote the experimented dual-encoder versions as $s - p - b - h - \alpha$ where $s \in \{1, 2\}$ is the ordinal number of the training stage, $p \in \{0, 3\}$ is the number of epochs that the PhoBERT encoders were fine-tuned with the MLM task on the law corpus before fine-tuning on the question-passage dataset in the first stage of dual-encoder training (in the second stage or $s = 2$, we start from the resulted encoder in the first stage, so $p = 0$), $b \in \{16, 32, 64\}$ is the batch size, $h \in \{0, 1, 3\}$ is the number of hard negatives used for each training sample, and $\alpha \in [0, 1]$ is the hyperparameter used in the loss function.

Table 1. Dual-encoder: Hard negatives experiments.

$\mathbf{s - p - b - h - \alpha}$	$\mathbf{acc_1}$	$\mathbf{acc_5}$	$\mathbf{acc_{10}}$	$\mathbf{acc_{30}}$	$\mathbf{acc_{100}}$
$1 - 0 - 16 - 0 - 0$	56.5	83.1	88.5	93.4	**97.0**
$1 - 0 - 32 - 0 - 0$	54.2	80.7	87.2	**94.1**	96.1
$1 - 0 - 64 - 0 - 0$	53.6	81.6	86.7	**94.1**	96.6
$1 - 0 - 16 - 1 - 1$	56.5	79.8	85.2	92.8	96.1
$1 - 0 - 32 - 1 - 1$	56.5	82.0	87.4	**94.1**	96.6
$1 - 0 - 16 - 3 - 1$	60.0	80.0	85.4	92.6	96.4
$1 - 0 - 16 - 3 - 0.1$	**63.9**	**84.0**	**89.6**	**94.1**	**97.0**

Table 1 represents our first experimental results in training a dual-encoder model, with different usages of hard-negative. **Table 1** shows that the use of hard negatives does not improve and even worsens the performances, which is contrary to previous studies [7, 13, 18]. After proposing a new loss function, we try it with different values of α. With $\alpha = 0.1$, we obtain the best performance that surpasses all previous results.

Table 2 represents our extremely impressive performance after fine-tuning PhoBERT for 3 epochs with the MLM task. This performance is even higher than when the new modified loss is applied. Using both fine-tuning and the new loss (with $\alpha = 0.1$) in the first training stage brings us the best result for the dual-encoder in the first stage of training.

Training the dual-encoder in the second stage does benefit the model performance, with higher results in the top 1, 5, 10, 30 (**Table 2**). Besides, using the new loss function continues to show its effect that when $\alpha = 0.3$, we obtain equal or higher accuracy on top 1, 10, 30, 100 retrieved passages than using the old loss function ($\alpha = 1$).

Table 2. Dual-encoder: Multi-stage training results.

$\mathbf{s - p - b - h - \alpha}$	$\mathbf{acc_1}$	$\mathbf{acc_5}$	$\mathbf{acc_{10}}$	$\mathbf{acc_{30}}$	$\mathbf{acc_{100}}$
$1 - 0 - 16 - 3 - 1$	60.0	80.0	85.4	92.6	96.4
$1 - 0 - 16 - 3 - 0.1$	63.9	84.0	89.6	94.1	97.0
$1 - 3 - 16 - 3 - 1$	64.3	84.7	89.4	94.8	**98.4**
$1 - 3 - 16 - 3 - 0.1$	65.4	86.7	91.9	96.4	**98.4**
$2 - 0 - 16 - 3 - 1$	70.6	**91.0**	**93.7**	97.0	97.9
$2 - 0 - 16 - 3 - 0.3$	**73.0**	88.7	**93.7**	**97.3**	**98.4**

Table 3 shows the results of the whole retriever model including the dual-encoder and cross-encoder re-ranker. Truncating long articles into smaller sub-passages that help PhoBERT encode the entire content can improve the dual-encoder performance. When using a cross-encoder and re-rank the top 30 retrieved passages with the combined score introduced in Section 3, the results increase significantly, stating the importance of a re-ranker after the dual-encoder in a retrieval system.

Table 3. Results when using cross-encoder re-ranker.

$\mathbf{s - p - b - h - \alpha}$	**trunc**	**re**	$\mathbf{acc_1}$	$\mathbf{acc_5}$	$\mathbf{acc_{10}}$	$\mathbf{acc_{30}}$	$\mathbf{acc_{10(}}$
$2 - 0 - 16 - 3 - 0.3$	No	No	73.0	88.7	93.7	**97.3**	**98.4**
$2 - 0 - 16 - 3 - 0.3$	Yes	No	73.5	90.1	92.6	96.8	**98.4**
$2 - 0 - 16 - 3 - 0.3$	Yes	Yes	**83.4**	**95.2**	**96.4**	96.8	-

Dataset Limitation. After analyzing the cases where the retriever model failed, we found that the main reason can be mentioned is the questions that require high inference. However, at the same time, there is a problem caused by the lack of relevant information on the positive passages to the answer. The corpus only includes legal passages taken from a large legal document and omits the title of the whole legal document. We find that the title of a legal document can also be an important source of information that improves the quality of the model. Therefore, we crawl[5] the titles of legal documents in the corpus to attach to each passage and re-train both the dual-encoder and cross-encoder. For documents with too short titles, we have also added the content of *Article 1: Scope of Regulation* to increase the amount of information for the passages. The results showed that adding text titles to the rules improved the model's results significantly.

[5] From the website: https://thuvienphapluat.vn/

Table 4. Dual-encoder: Stage 1 results with titles appending.

$\mathbf{s-p-b-h-\alpha}$	**title**	$\mathbf{acc_1}$	$\mathbf{acc_5}$	$\mathbf{acc_{10}}$	$\mathbf{acc_{30}}$	$\mathbf{acc_{100}}$
$1-3-16-3-1$	No	64.3	84.7	89.4	94.8	**98.4**
$1-3-16-3-0.1$	No	65.4	86.7	91.9	96.4	**98.4**
$1-3-16-3-1$	Yes	66.3	86.9	91.2	**96.6**	98.2
$1-3-16-3-0.1$	Yes	**69.2**	**89.0**	**93.2**	96.4	**98.4**

According to **Table 4** and **Table 5**, appending law titles does improve the performance of the dual-encoder in both training stages. When we use the old loss function ($\alpha = 1$), the results with attached passages can approximate the highest performance with the modified loss and original articles. Combining the modified loss and titles-appending brings us very comparative results. After two-stage training, the dual-encoder can retrieve around 74.4% of the questions with one returned passage.

Table 5. Dual-encoder: Stage 2 results with titles appending.

$\mathbf{s-p-b-h-\alpha}$	**title**	$\mathbf{acc_1}$	$\mathbf{acc_5}$	$\mathbf{acc_{10}}$	$\mathbf{acc_{30}}$	$\mathbf{acc_{100}}$
$2-0-16-3-1$	No	70.6	91.0	93.7	97.0	97.5
$2-0-16-3-0.3$	No	73.0	88.7	93.7	97.3	97.9
$2-0-16-3-1$	Yes	73.5	**91.9**	93.4	96.8	98.2
$2-0-16-3-0.3$	Yes	**74.4**	91.4	**94.1**	**97.3**	**98.4**

After truncating and re-ranking, our new retriever surpasses the previous on all top 1, 5, 10 re-ranked retrieved passages.

Table 6. Re-ranker results with titles appending.

$\mathbf{s-p-b-h-\alpha}$	**title**	**trunc**	**re**	$\mathbf{acc_1}$	$\mathbf{acc_5}$	$\mathbf{acc_{10}}$	$\mathbf{acc_{30}}$
$2-0-16-3-0.3$	No	Yes	Yes	83.4	95.2	96.4	96.8
$2-0-16-3-0.3$	Yes	Yes	Yes	**84.3**	**95.7**	**96.6**	**97.3**

5 Conclusion

In this paper, we introduce a training strategy to improve the performance of the retriever on closed-domain datasets despite facing limited parallel computing resources for the training procedure. By demonstrating the effectiveness of the techniques through experiments, we conclude that fine-tuning the pre-trained encoder on the passages corpus and applying a modified loss in a multi-stage training procedure is the right approach for the dual-encoder. The dual-encoder should be combined with a cross-encoder to form a final robust retriever, which shows potential to advance the state-of-the-art in question-answering systems, especially on specific domains.

References

1. Chang, W.C., Yu, F.X., Chang, Y.W., Yang, Y., Kumar, S.: Pre-training tasks for embedding-based large-scale retrieval (2020).
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2019).
3. Fajcik, M., Docekal, M., Ondrej, K., Smrz, P.: R2-d2: A modular baseline for open-domain question answering (2021).
4. Gao, L., Callan, J.: Condenser: a pre-training architecture for dense retrieval (2021).
5. Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M.W.: Realm: Retrieval-augmented language model pre-training (2020).
6. Johnson, J., Douze, M., J´egou, H.: Billion-scale similarity search with gpus (2017).
7. Karpukhin, V., O˘guz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., tau Yih, W.: Dense passage retrieval for open-domain question answering (2020).
8. Lee, K., Chang, M.W., Toutanova, K.: Latent retrieval for weakly supervised open domain question answering (2019).
9. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach (2019).
10. Luan, Y., Eisenstein, J., Toutanova, K., Collins, M.: Sparse, dense, and attentional representations for text retrieval (2021).
11. Nguyen, D.Q., Nguyen, A.T.: Phobert: Pre-trained language models for Vietnamese(2020).
12. Nogueira, R., Cho, K.: Passage re-ranking with BERT (2020).
13. Qu, Y., Ding, Y., Liu, J., Liu, K., Ren, R., Zhao, W.X., Dong, D., Wu, H., Wang, H.: RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering (2021).
14. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks (2019).
15. Robertson, S., Walker, S.: Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. pp. 232–241 (01 1994)..https://doi.org/10.1007/978-1-4471-2099-524
16. Robertson, S.E., Zaragoza, H.: The probabilistic relevance framework: Bm25 and beyond. Found. Trends Inf. Retr. 3, 333–389 (2009).
17. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manag. 24, 513–523 (1988).
18. Xiong, L., Xiong, C., Li, Y., Tang, K.F., Liu, J., Bennett, P., Ahmed, J., Over-wijk, A.: Approximate nearest neighbor negative contrastive learning for dense text retrieval (2020)
19. Zhan, J., Mao, J., Liu, Y., Guo, J., Zhang, M., Ma, S.: Optimizing dense retrieval model training with hard negatives (2021).