Learning Discrete Document Representations in Web Search

Rong Huang*
Peking University
aroon@stu.pku.edu.cn

Danfeng Zhang*
Baidu Inc.
f592375753@gmail.com

Weixue Lu
Baidu Inc.
luweixue@baidu.com

Han Li
Meng Wang
Daiting Shi
Baidu Inc.
lihan08@baidu.com
wangmeng12@baidu.com
shidaiting01@baidu.com

Jun Fan
Zhicong Cheng
Simiu Gu
Baidu Inc.
fanjun@baidu.com
chengzhicong01@baidu.com
gusimiu@baidu.com

Dawei Yin ✉
Baidu Inc.
yindawei@acm.org

ABSTRACT

Product quantization (PQ) has been usually applied to dense retrieval (DR) of documents thanks to its competitive time, memory efficiency and compatibility with other approximate nearest search (ANN) methods. Originally, PQ was learned to minimize the reconstruction loss, i.e., the distortions between the original dense embeddings and the reconstructed embeddings after quantization. Unfortunately, such an objective is inconsistent with the goal of selecting ground-truth documents for the input query, which may cause a severe loss of retrieval quality. Recent research has primarily concentrated on jointly training the biencoders and PQ to ensure consistency for improved performance. However, it is still difficult to design an approach that can cope with challenges like discrete representation collapse, mining informative negatives, and deploying effective embedding-based retrieval (EBR) systems in a real search engine.

In this paper, we propose a Two-stage Multi-task Joint training technique (TMJ) to learn discrete document representations, which is simple and effective for real-world practical applications. In the first stage, the PQ centroid embeddings are regularized by the dense retrieval loss, which ensures the distinguishability of the quantized vectors and preserves the retrieval quality of dense embeddings. In the second stage, a PQ-oriented sample mining strategy is introduced to explore more informative negatives and further improve the performance. Offline evaluations are performed on a public benchmark (MS MARCO) and two private real-world web search datasets, where our method notably outperforms the SOTA PQ methods both in Recall and Mean Reciprocal Ranking (MRR). Besides, online experiments are conducted to validate that our technique can significantly provide high-quality vector quantization. Moreover, our joint training framework has been successfully applied to a billion-scale web search system.

CCS CONCEPTS

• **Information systems** → **Novelty in information retrieval**.

KEYWORDS

PQ, Joint Training, Web Search Engine, Negative Sample Mining

ACM Reference Format:
Rong Huang[1], Danfeng Zhang[1], Weixue Lu, Han Li, Meng Wang, Daiting Shi, Jun Fan, Zhicong Cheng, Simiu Gu, and Dawei Yin ✉. 2023. Learning Discrete Document Representations in Web Search. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3580305.3599854

1 INTRODUCTION

Due to significant development in representation learning [1] and deep pretrained language models [5, 26, 34], the dense retrieval (DR) [15, 25] has become a prominent paradigm for improving retrieval performance. In this paradigm, biencoders [12, 16, 20] are employed to embed user queries and documents in a latent vector space, and relevant documents are selected from the entire corpus based on embedding similarity. Since the documents must be retrieved from a large-scale corpus, a brute-force linear scan is temporally infeasible. Instead, approximate nearest neighbour search (ANN) [23] is employed to support real-world information retrieval, which can efficiently identify documents with high embedding similarities. For the past decade, vector compression techniques have been widely applied, with Product quantization (PQ) [10, 18] being one of the most prominent, enabling ANN to be performed with competitive time and memory efficiency.

Typical learning paradigms of PQ. PQ is typically learned to minimize the reconstruction loss, which refers to reducing distortions (e.g., $l2$ distance) between the original dense embeddings and the reconstructed embeddings from quantization. Unfortunately, such an operation is inconsistent with the goal of retrieving ground-truth documents for the query, which may cause severe loss of retrieval performance. Recent works [3, 13, 35–37, 39, 41, 42, 44] have mostly focused on jointly training the biencoders and PQ to

✉Corresponding author.

*Equal contribution.

KDD '23, August 6–10, 2023, Long Beach, CA, USA.

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599854

ensure consistency for improved performance. Nevertheless, few works have coped with the following challenges all at once:

- **Discrete representation collapse**. Direct PQ centroid optimization produces indistinguishable quantized vectors, which will incur a significant reduction in retrieval quality [35, 41]. Thus, previous works have been devoted to elaborating fine regularizations to generate more balanced clustering distribution. However, as we will see later in our experiments, all these methods will greatly hurt the retrieval effectiveness of the dense embeddings before quantization, which we believe is critical for industrial applications in terms of cost and performance preservation.
- **Mining informative negatives**. Hard negatives are important for PQ learning [36, 41, 45], which helps to achieve faster convergency and better optimization. However, the existing joint learning methods either simply treat all top retrieved non-ground-truth results equally as negatives or focused on detecting and reweighting false negatives. Few studies have attempted to mine harder negative samples by fully utilizing the discrete document representations after PQ learning.
- **Billion-scale industrial deployment**. To create significant impact to real-world applications, it also calls for practical solutions on elaborating and deploying fast and effective ANN systems to serve web-scale data, which must have competitive time and memory efficiency, as well as high-quality retrieval performance.

Present work. In this paper, to address the above challenges, we propose a Two-stage Multi-task Joint training technique (TMJ) to learn discrete document representations, which is simple and effective for real-world practical applications. In the first stage, learning targets based on PQ are regularized by the dense retrieval loss, which ensures the distinguishability of the quantized vectors and preserves the retrieval quality of dense embeddings. Specifically, we combine three losses, namely 1) the reconstruction loss, 2) the retrieval loss based on quantized embeddings, and 3) the retrieval loss based on dense embeddings, and train PQ in a multi-task way using linear combination of losses. The first two losses aim to improve clustering qualities and retrieval effectiveness of quantized vectors. Nonetheless, we discovered that merely addressing these two losses will greatly hurt the retrieval effectiveness of dense embeddings due to the aforementioned representation collapse problem, while the third loss yields more distinguishable embeddings. This contradiction regularizes the model to produce more even distribution of PQ centroids. Besides, we believe that a reasonable joint learning strategy should not only focus on the recall effectiveness of the compressed vectors, but should also corrupt the original dense embeddings as little as possible and largely maintain their performance. The integration of the third loss contributes to achieving these goals. In addition, due to the slight violation of dense vectors, the query embeddings can only be predicted once in real applications, and serve both the retrieval process based on PQ and the subsequent reranking process based on dense vectors. In the second stage, we introduce a PQ-oriented sample mining strategy to explore more informative negatives. For each query, we consider top retrieved non-ground-truth results by both the discrete and dense document vectors to be hard negatives. In this way, it further improves the retrieval performance of discrete document representations.

To verify the effectiveness of our method, extensive offline evaluations are performed on a public benchmark (MS MARCO) and two private real-world web search datasets, where our method significantly outperforms the SOTA PQ methods both in Recall and MRR. Furthermore, the ablation study demonstrates the necessity of our two-stage multi-task design.

Moreover, we develop an effective and efficient ANN system for large-scale web retrieval. Proximity graph methods, such as HNSW [30], are considered the current state-of-the-art for ANN search because of their high recall rate and low recall latency. To the best of our knowledge, few practices have been attempted to apply jointly learned PQ to an HNSW-based billion-scale information retrieval system. By doing so, online experiments demonstrate that our proposed technique is able to significantly provide high-quality vector quantization in terms of online DCG evaluation and manual side-by-side comparison.

To summarize, our work is highlighted with the following points: 1) We propose a novel and effective two-stage multi-task joint training technique to learn discrete document representations. 2) A PQ-oriented sample mining strategy is introduced to explore more informative negatives, which further improves quantization quality. 3) The jointly learned PQ is successfully applied to an HNSW-based billion-scale information retrieval system. 4) Extensive offline and online experiments verify the effectiveness of our method.

2 RELATED WORK

PQ and joint training. Product quantization [18] is a popular vector compression method to enable efficient and effective ANN search, where discrete document representations are learned and encoded into compact indexes. The early PQ methods [10, 18] are unsupervised, which takes the well-trained output embeddings of the encoders as input and learns the quantization model with heuristic algorithms (e.g., k-means). Recently, encoders and quantization models are jointly trained based on supervised query-document pairs, and these methods [2, 3, 9, 21, 41, 42] can be collectively referred to as supervised PQ. To achieve more effective exploitation of massive unlabeled data, Xiao et al. [36] propose Distill-VQ to learn from dense embeddings via knowledge distillation. However, we find that the existing joint training methods improve PQ retrieval effectiveness but weakens the performance of dense vectors generated by encoders (see Section 3.3 and 5.2.2). To tackle this problem, we introduce the multi-task loss in the training process, which retains the advantages of joint training while making the dense vectors more distinguishable.

Negative sampling. Hard negatives may be roughly categorized into static hard negatives and dynamic hard negatives depending on whether the negative selection is fixed or updated. Zhan et al. [40] shows that using only static hard negatives [7, 14, 20, 28, 38] may cause unstable training and cannot guarantee reliable performance. Some methods [39, 41] employ dynamic hard negatives, the top irrelevant documents retrieved at each training step, to improve the retrieval quality. Besides, many works [33, 46] have focused on alleviating the false negative problems while applying hard negatives. In this paper, we propose a PQ-oriented negative mining strategy, which fully exploits the retrieval gap between the quantized and dense embeddings.

Industrial deployment. Representation-based models with an ANN algorithm have become the mainstream trend to efficiently deploy neural retrieval models in industry. Taobao [22], Amazon [31] and JD [43] built their respective EBR systems in their e-commerce search engines. For social network, Facebook introduced a unified embedding framework developed to model semantic embeddings [15]. In recent years, some studies, e.g., DiskANN [17], have focused on time and memory scalable ANN and proposed solutions based on graph index. For sponsored search, Bing developed Uni-Retriever [44], a unified learning framework using ad-hoc compression methods to efficiently retrieve high-relevance and high-CTR ads. Knowing that time cost, memory usage and retrieval quality all remain great challenges for billion-scale information retrieval system, we substantially optimize the compressed embeddings and deploy fast and effective ANN systems. Related technical discussions and experimental studies following may provide insightful information for real-world applications.

3 METHODOLOGY

In this section, we propose TMJ, which stands for a two-stage multi-task joint training technique for learning discrete document representations in web search systems. We first discuss Product Quantization preliminaries in Section 3.1. Then, we provide an overview of the proposed framework in Section 3.2. Finally, we present our multi-task loss and PQ-oriented negative mining strategy in Section 3.3 and Section 3.4, respectively.

3.1 Preliminaries

PQ is a kind of classical and effective vector compression method, especially suitable for very large databases. For further discussion, we define and use the similar mathematical symbols as Zhan et al. [41]. Let $\boldsymbol{d} \in \mathbb{R}^D$ denote the uncompressed document embedding (also called unquantized embedding or original dense embedding) generated by the biencoder, and $\hat{\boldsymbol{d}} \in \mathbb{R}^D$ denote the compressed document embbeding (also known as quantized embeddings). For vectors of dimension D, PQ is parameterized by the PQ Centroid Embeddings, which are defined as M sets of embeddings and each set includes K embeddings of dimension D/M. Formally, let $\boldsymbol{c}_{i,j}$ be the j_{th} centroid embedding from i_{th} sub-vector set:

$$\boldsymbol{c}_{i,j} \in \mathbb{R}^{\frac{D}{M}}, 1 \le i \le M, 1 \le j \le K.$$

At inference time, PQ first splits the uncompressed vector $\boldsymbol{d}$ equally into M sub-vectors of dimension $\frac{D}{M}$, i.e.,

$$\boldsymbol{d} = \text{concat}[\boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_M]. \tag{1}$$

Then PQ independently quantizes each sub-vector to the nearest PQ Centroid Embedding. Formally, to quantize a sub-vector $\boldsymbol{d}_i$, PQ selects the $\gamma_i(\boldsymbol{d})_{\text{th}}$ centroid embedding, which achieves the minimum quantization error, from the i_{th} centroid set:

$$\gamma_i(\boldsymbol{d}) = \arg\min_j \left\| \boldsymbol{c}_{i,j} - \boldsymbol{d}_i \right\|^2. \tag{2}$$

The M selected centroid embeddings are then concatenated and reconstructed as the quantized document embedding $\hat{\boldsymbol{d}}$:

$$\hat{\boldsymbol{d}} = \text{concat}\left[\boldsymbol{c}_{1,\gamma_1(\boldsymbol{d})}, \boldsymbol{c}_{2,\gamma_2(\boldsymbol{d})}, ..., \boldsymbol{c}_{M,\gamma_M(\boldsymbol{d})}\right]. \tag{3}$$

During the training process of PQ, parameters $\{\boldsymbol{c}_{i,j}\}$ are trained to minimize the MSE loss between the original document embedding and the quantized one:

$$L_m = \left\| \text{sg}(\boldsymbol{d}) - \hat{\boldsymbol{d}} \right\|^2, \tag{4}$$

where, similar as previous works [3, 42], $\text{sg}(\cdot)$ is the stop gradient operator, which is identity function in forward pass, but drops gradient for variables inside it during the backward pass. The loss of Eq. 4 is also called the reconstruction loss. In this way, the vectors before and after quantization are close to each other to preserve the retrieval quality of dense embeddings.

PQ enables ANN to be performed with competitive time and memory efficiency. For each sub-vector of the quantized embedding, we only need to preserve the centroid ID which can be represented with much lower memory cost. In addition, during online services, the inner product between subvectors of the query and PQ centroid embeddings are precomputed and stored into a lookup table. Hence the complexity of distance computations can be largely reduced by replacing a majority of arithmetic operations with table lookups.

3.2 Overview

Figure 1 provides an illustration of the proposed framework. The framework can be organized into two stages.

In the first stage, we jointly train the biencoder and PQ centroids with our multi-task loss (see Eq. 9). During training, for each query q_i, the hard negative documents n_{ik} $(1 \le k \le j)$ are sampled from top irrelevant documents pre-retrieved by the warm-up model with initialized PQ parameters.

In the second stage, following RepCONC [41], we further train the query encoder and PQ parameters by minimizing the retrieval loss w.r.t the quantized embeddings (see Eq. 6) with our proposed PQ-oriented negative mining strategy. For each query q_i, we take the irrelevant documents commonly retrieved by the PQ index (constructed based on the quantized embeddings) and Flat index (constructed based on the dense embeddings) as hard negatives n_{ik} $(1 \le k \le j)$, which are informative for both retrieval quality improvement and PQ learning.

3.3 Multi-task Loss

In this section, we discuss the weaknesses of the standard joint training objective of previous studies and show that it harms the retrieval performance of uncompressed embeddings generated by the biencoder. At the same time, we state the necessity of our multi-task loss. Formally, the commonly used retrieval loss based on uncompressed embeddings (or dense embeddings) is formulated as:

$$L_r = -\ln \frac{e^{\langle \boldsymbol{q}, \boldsymbol{d}^+ \rangle}}{e^{\langle \boldsymbol{q}, \boldsymbol{d}^+ \rangle} + \sum_{\boldsymbol{d}^-} e^{\langle \boldsymbol{q}, \boldsymbol{d}^- \rangle}}, \tag{5}$$

where $\boldsymbol{d}^+$ and $\boldsymbol{d}^-$ are relevant and irrelevant documents respectively, $\langle \cdot, \cdot \rangle$ measures the similarity between queries and documents. L_r facilitates effective representation learning by encouraging the relevant pairs to be closer than irrelevant pairs in the embedded

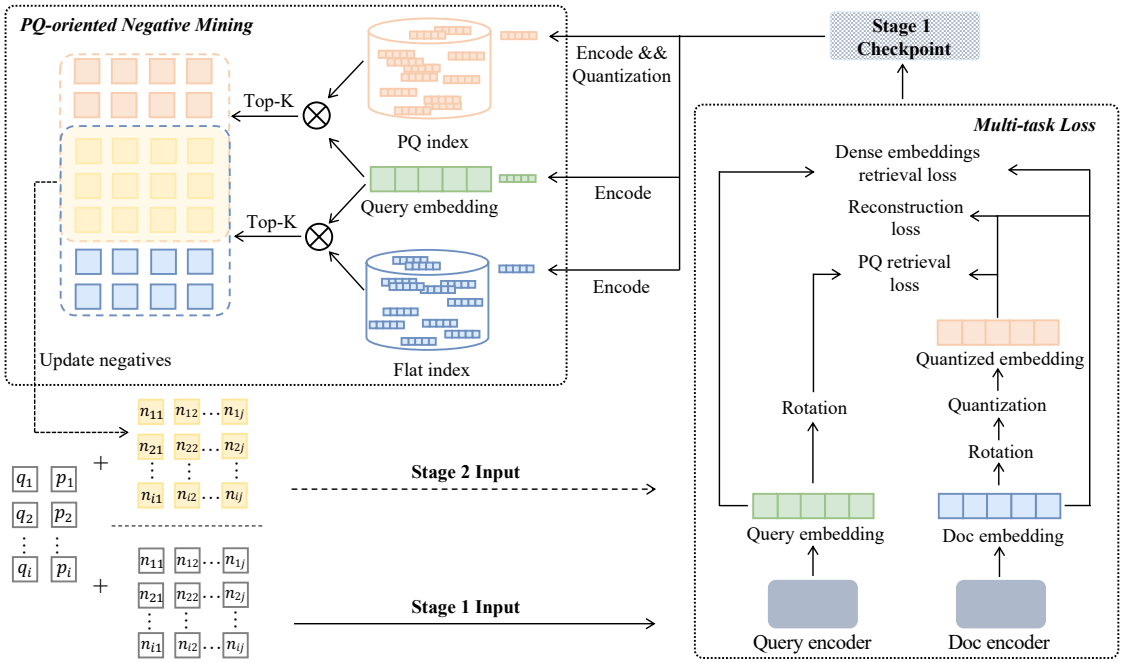

Figure 1: Training framework of TMJ.

vector space. Similarly, the retrieval loss based on compressed embeddings (or based on PQ) can be written as:

$$L_{\hat{r}} = -\ln \frac{e^{\langle \boldsymbol{q}, \hat{\boldsymbol{d}}^+ \rangle}}{e^{\langle \boldsymbol{q}, \hat{\boldsymbol{d}}^+ \rangle} + \sum_{\hat{\boldsymbol{d}}^-} e^{\langle \boldsymbol{q}, \hat{\boldsymbol{d}}^- \rangle}}. \tag{6}$$

Since the arg min operation in Eq. 2 is non-differentiable, straightforward optimization (by the standard back propagation algorithm) of $L_{\hat{r}}$ with respect to the biencoder parameters is infeasible. To enable a pseudo gradient, we leverage the gradient straight-through estimator by adjusting the original quantization as follows:

$$\tilde{\boldsymbol{d}} = \boldsymbol{d} - \mathrm{sg}(\boldsymbol{d} - \hat{\boldsymbol{d}}), \tag{7}$$

where $\mathrm{sg}(\cdot)$ is again the stop gradient operation. From Eq. 7, we have $\tilde{\boldsymbol{d}} = \hat{\boldsymbol{d}}$ in the forward pass, $\partial \tilde{\boldsymbol{d}} = \partial \boldsymbol{d}$ during the backward pass.

Previous methods [39, 41, 42] for joint training the model and index primarily relied on the retrieval loss based on PQ and the reconstruction loss. By combining L_m with $L_{\hat{r}}$, the conventional loss of joint training is:

$$L_0 = L_m + L_{\hat{r}}. \tag{8}$$

However, we find that L_0 does damage to the original representation learning of the biencoder, i.e., the minization of L_r. This is demonstrated by the experimental result, which will be shown in Table 2, that the retrieval performance of the original dense embeddings is significantly decreased after joint training. Thus, instead of optimizing L_0, we propose a multi-task loss by adding L_r to the conventional loss of joint training:

$$L_1 = L_m + L_{\hat{r}} + L_r. \tag{9}$$

The motivation for the design is: 1) to preserve the retrieval effectiveness of the original dense embeddings, which we believe a good quantization algorithm should be further finetuning based on the well-trained warm-up model and should not violate the dense embeddings too much; 2) to generate more balanced clustering distribution since L_r makes the dense embeddings more distinguishable.

To observe the effect of our multi-task loss on embeddings, we assume that the training and test sets are identically distributed, over which we use $\mathbf{E}[L]$ to denote the expectation of loss value L. Then, let $\boldsymbol{d}^*$ and $\boldsymbol{d}^{**}$ be the dense embeddings generated by the model that minimizes $\mathbf{E}[L_0]$ and $\mathbf{E}[L_1]$, respectively, on the training set. Let $\hat{\boldsymbol{d}}^*$ and $\hat{\boldsymbol{d}}^{**}$ denote the associated quantized embeddings. Further, we use $L(\boldsymbol{d})$ to denote the loss value computed with $\boldsymbol{d}$, for example, $L_r(\boldsymbol{d}^*) = -\ln \frac{e^{\langle \boldsymbol{q}, \boldsymbol{d}^{*+} \rangle}}{e^{\langle \boldsymbol{q}, \boldsymbol{d}^{*+} \rangle} + \sum_{\boldsymbol{d}^{*-}} e^{\langle \boldsymbol{q}, \boldsymbol{d}^{*-} \rangle}}$. Hence, it follows from the definition of minimization that

$$\mathbf{E}\left[L_0(\boldsymbol{d}^*)\right] \le \mathbf{E}\left[L_0(\boldsymbol{d}^{**})\right], \tag{10}$$

$$\mathbf{E}\left[L_1(\boldsymbol{d}^{**})\right] \le \mathbf{E}\left[L_1(\boldsymbol{d}^*)\right]. \tag{11}$$

These imply that

$$\begin{aligned} &\mathbf{E}\left[L_r(\boldsymbol{d}^*)\right] - \mathbf{E}\left[L_r(\boldsymbol{d}^{**})\right] \\ &= \mathbf{E}\left[L_1(\boldsymbol{d}^*) - L_0(\boldsymbol{d}^*) - \left(L_1(\boldsymbol{d}^{**}) - L_0(\boldsymbol{d}^{**})\right)\right] \\ &= \mathbf{E}\left[L_1(\boldsymbol{d}^*) - L_1(\boldsymbol{d}^{**})\right] + \mathbf{E}\left[L_0(\boldsymbol{d}^{**}) - L_0(\boldsymbol{d}^*)\right] \\ &\ge 0, \end{aligned} \tag{12}$$

where the last inequality follows from Eq. 10 and 11.

Usually, $\mathbf{E}[L_r(\boldsymbol{d})]$ measures the retrieval quality of the uncompressed embeddings, where the lower value means better performance. From Eq. 12 and our experimental results in Table 2, it can be concluded that the uncompressed embeddings learned by minimizing our multi-task loss performs far better than those learned by minimizing the conventional joint training loss.

Here, we provide an intuitive explanation of our multi-task loss. The first term L_m optimizes the compressed embeddings by reducing quantization error. The second term $L_{\hat{r}}$ optimizes the compressed embeddings by focusing on the retrieval performance. It also indirectly influences the original uncompressed embeddings through gradient back propagation. By combining L_m and $L_{\hat{r}}$, the conventional loss L_0 requires the uncompressed embeddings and the commonly-shared quantization centroids to be closer, which enforces compact clustering of the embeddings but decreases their retrieval performance. As a result, by adding the third term L_r, our multi-task loss can guarantee high retrieval performances for both uncompressed and compressed embeddings.

3.4 PQ-oriented Negative Mining

The key challenge associated with dense retrieval is to pick proper negative samples during its training process [20]. In addition, for the joint training of encoders and indexes, the selection of negative samples also has a significant impact on the final retrieval quality of compressed or uncompressed embeddings. Although a variety of negative sampling approaches [24, 32, 38] have been explored, there is still a lack of investigation on negative sampling for joint learning with respect to PQ.

Next, we describe our data mining method. For convenience of analysis, let us define some symbols. The document set retrieved by the uncompressed and compressed embeddings is denoted by S_u and S_c, respectively. The intersection of S_u and S_c is denoted by S_i, i.e., $S_i = S_u \cap S_c$. The set difference between S_u and S_c is denoted by S_d, i.e., $S_d = S_u - S_d$. Typically, traditional negative mining methods simply take the irrelevant top-k documents, retrieved by the uncompressed or compressed embeddings, as hard negatives. However, we find that S_i is more beneficial in the joint training process, since S_i is difficult and informative for both the biencoder and PQ index. This has also been verified in our experimental results (see Table 4).

Then, the mined negatives are applied to train $L_{\hat{r}}$ in the second stage of joint training. In order to observe the impact of these negative samples on joint training, we formally explain the effectiveness of S_i through the gradient norms of loss functions with respect to training samples [38]. First, through direct calculation, it can be obtained that the gradient of Eq. 6 is:

$$\begin{aligned}\frac{\partial L_{\hat{r}}}{\partial \hat{\boldsymbol{d}}^+} &= -\boldsymbol{q}\cdot\left(1-\frac{e^{\langle \boldsymbol{q},\hat{\boldsymbol{d}}^+\rangle}}{e^{\langle \boldsymbol{q},\hat{\boldsymbol{d}}^+\rangle}+\sum_{\hat{\boldsymbol{d}}^-} e^{\langle \boldsymbol{q},\hat{\boldsymbol{d}}^-\rangle}}\right)\\ &= -\boldsymbol{q}\cdot\left(1-\frac{1}{1+\sum_{\hat{\boldsymbol{d}}^-}\frac{e^{\langle \boldsymbol{q},\hat{\boldsymbol{d}}^-\rangle}}{e^{\langle \boldsymbol{q},\hat{\boldsymbol{d}}^+\rangle}}}\right)\end{aligned} \tag{13}$$

Assume that all embeddings have been normalized in advance, for example, $\|\boldsymbol{q}\|^2 = 1$. Then, the gradient norm with respect to the positive sample in Eq. 13 is:

$$\left\|\frac{\partial L_{\hat{r}}}{\partial \hat{\boldsymbol{d}}^+}\right\| = 1-\frac{1}{1+\sum_{\hat{\boldsymbol{d}}^-}\frac{e^{\langle \boldsymbol{q},\hat{\boldsymbol{d}}^-\rangle}}{e^{\langle \boldsymbol{q},\hat{\boldsymbol{d}}^+\rangle}}}, \tag{14}$$

where $\|\cdot\|$ is the L^2-norm operation. For the convenience of more detailed analysis, we define the set of negative samples, in which the positive has higher similarity to the given query $\boldsymbol{q}$, i.e.,

$$S^1 = \left\{\hat{\boldsymbol{d}}^- \in \mathbb{R}^D \middle| \delta > 0, \delta = \langle \boldsymbol{q},\hat{\boldsymbol{d}}^+\rangle - \langle \boldsymbol{q},\hat{\boldsymbol{d}}^-\rangle\right\}. \tag{15}$$

Similarly, the set of negative samples, in which the positive has lower similarity to the query is defined as:

$$S^2 = \left\{\hat{\boldsymbol{d}}^- \in \mathbb{R}^D \middle| \delta \le 0, \delta = \langle \boldsymbol{q},\hat{\boldsymbol{d}}^+\rangle - \langle \boldsymbol{q},\hat{\boldsymbol{d}}^-\rangle\right\}. \tag{16}$$

Then, Eq. 14 can be rewritten as:

$$\begin{aligned}\left\|\frac{\partial L_{\hat{r}}}{\partial \hat{\boldsymbol{d}}^+}\right\| &= 1-\frac{1}{1+\sum_{\boldsymbol{d}^-\in S^1}\frac{e^{\langle \boldsymbol{q},\hat{\boldsymbol{d}}^-\rangle}}{e^{\langle \boldsymbol{q},\hat{\boldsymbol{d}}^+\rangle}}+\sum_{\boldsymbol{d}^-\in S^2}\frac{e^{\langle \boldsymbol{q},\hat{\boldsymbol{d}}^-\rangle}}{e^{\langle \boldsymbol{q},\hat{\boldsymbol{d}}^+\rangle}}}\\ &\ge 1-\frac{1}{1+|S^2|},\end{aligned} \tag{17}$$

where $|\cdot|$ stands for the operation of taking the count of elements in a set, and the last inequality holds due to the definition of S^2, i.e.,

$$\sum_{\boldsymbol{d}^-\in S^2}\frac{e^{\langle \boldsymbol{q},\hat{\boldsymbol{d}}^-\rangle}}{e^{\langle \boldsymbol{q},\hat{\boldsymbol{d}}^+\rangle}} = \sum_{\boldsymbol{d}^-\in S^2} e^{\langle \boldsymbol{q},\hat{\boldsymbol{d}}^-\rangle - \langle \boldsymbol{q},\hat{\boldsymbol{d}}^+\rangle} \ge |S^2|. \tag{18}$$

It can be clearly seen from Eq. 14 that $\left\|\frac{\partial L_{\hat{r}}}{\partial \hat{\boldsymbol{d}}^+}\right\|$ increases with the growth of $\Sigma_{\hat{\boldsymbol{d}}^-}\frac{e^{\langle \boldsymbol{q},\hat{\boldsymbol{d}}^-\rangle}}{e^{\langle \boldsymbol{q},\hat{\boldsymbol{d}}^+\rangle}}$. Besides, from Eq. 17, we can see that the lower bound of $\left\|\frac{\partial L_{\hat{r}}}{\partial \hat{\boldsymbol{d}}^+}\right\|$ varies directly with the number of elements in set S^2, where the negative samples are hard that the model fails to distinguish from the positive. Hence, when the number of elements in set S^2 is large, it has a positive impact on the retrieval performance because of the large gradient norm.

Returning to our PQ-oriented negative sampling method, the samples from S_i (the intersection of S_u and S_c) are more difficult, because they are incorrectly retrieved by both PQ index and dense index. From Eq. 14 and 17, harder negatives lead to larger gradient norms. As a result, compared with other negative sample methods, our method is more conducive to the representation learning of the biencoder and the training of PQ centroids.

4 INFERENCE AND SERVING

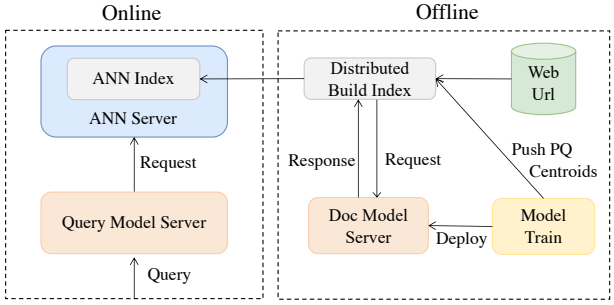

Figure 2: The searching system framework.

The important challenge with online dense retrieval is the trade-off between effectiveness and efficiency. Brute force algorithms that retrieve the exact closest ones of a query vector from large amounts of document vectors cannot satisfy the demand for low online time latency. There are different kinds of ANN search algorithms based on hash [11], graph [29] and so on. We apply HNSW [30] with TMJ

to our online system considering time and memory complexity. Our searching system is shown as Figure 2. It can be seen that our database is a huge resource of web urls. In the offline process, the index is built using the compressed document embeddings obtained from the doc model server, as well as the PQ centroids from the trained model. Given a query embedding, we compute and store the distances of its sub-vectors to PQ centroids, and obtain the retrieval results by summing up the precomputed distances.

5 EXPERIMENT

5.1 Experimental Setup

5.1.1 Datasets. We conduct main experiments on the passage retrieval task of the TREC 2020 Deep Learning Track [4]. The task has large training sets based on human relevance assessments, derived from MS MARCO. It extracts contextual passages from actual Web documents as answers, with all questions drawn from real anonymous user queries. There are a total of 8.8 million passages, half a million training queries, 7 thousand development queries and 54 test queries. We leverage official metrics to measure the retrieval quality, such as NDCG@10 and MRR@10. Besides, the recall rate is adopted for more comprehensive evaluation.

5.1.2 Baselines. We list two groups of compressed baselines in experiments including unsupervised methods and supervised methods. In the case of unsupervised methods, we use Faiss [19] to implement the baselines for PQ [18] and OPQ [10], and ScaNN [13] is implemented based on open source code. For supervised methods, we reproduce the latest works on the joint learning including JPQ [39], RepCONC [41] and Distill-VQ [36]. JPQ trains the query encoder and PQ index jointly in an end-to-end manner, and RepCONC further adopts uniform clustering constraint based on JPQ. Unlike the above two models, Distill-VQ unifies learning of IVF and PQ within a knowledge distillation framework. To make a fair comparison, we reproduce the results of Distill-VQ without IVF module during training and inference, and all methods do not rerank the candidates after retrieval.

5.1.3 Implementation details. Note that although joint training substantially improves the retrieval performance of Product Quantization, there are inevitable losses compared with dense embeddings retrieval. Learning of compressed vectors based on a good retrieval model is more reasonable. Therefore, all models use CoCondenser [6], which is the latest released well-trained encoder on MS MARCO, as the dense encoder during experiments. Since it does not compress the index, it is an ideal upper bound for the compressed index. Following ADORE [40], training consists of two stages. We use multi-target loss in first stage, and replace it with $L_{\hat{r}}$ in second stage. In first stage, learning rates are 2×10^{-5} and 5×10^{-4} separately for encoders and centroid embeddings. In second stage, the doc encoder is fixed but query encoder is kept updated, and learning rates are 2×10^{-6} and 2×10^{-5} separately for encoder and centroid embeddings. The embedding dimension D is 768 and the similarity function is inner product. The number of sub-vectors M is set to 24 and the number of centroids of each set K is set to 256. The compression ratio for all compression methods is set at 128x, which is equal to 4D/M. As for hardware, we use 8*NVIDIA-A100-40GB GPUs for training and inference. Besides, we optimize the parameters with the AdamW optimizer [27] and Gradient Cache [8].

5.2 Experiment Analysis

5.2.1 Comparison with compression methods. The overall results are presented in Table 1. To better illustrate the compression effect, we show the performance of the initialized model CoCondenser in the first row of the table, which does not use compression methods at all and just optimizes for the quality of the dense retrieval. The table's middle section presents performance for unsupervised methods, while the bottom section reports the performance of supervised methods. It can be observed that the performances of supervised approaches are generally better than the performances of unsupervised ones. Among the unsupervised methods, OPQ significantly outperforms vanilla PQ and ScaNN. The goal of all supervised methods is to optimize retrieval performance rather than simply minimizing reconstruction loss. Distill-VQ can leverage unlabeled data but relies on well-trained uncompressed embeddings because of knowledge distillation framework. JPQ uses fixed Index Assignments generated by OPQ, which cannot be updated during the training process. RepCONC derives an approximate solution for constrained clustering while focus less on the quality of the uncompressed vector. By introducing multi-task loss and PQ-oriented negative mining, our model effectively balances the quality of compressed and uncompressed vectors. Our proposed method achieves 0.3568 on MRR@10 and 0.6673 on NDCG@10, outperforming the strongest baselines by +0.71% and +1.32%, respectively.

5.2.2 Impact of loss function. Before the popularity of joint learning, previous methods [10, 13, 18] divide encoding and compression into two independent parts, which prevents the models from optimizing the PQ index using the supervised information. Therefore, the latest works [36, 39, 41] leverage PQ retrieval loss to solve this problem. Although representation learning encourages vectors to be distinguishable, clustering encourages vectors to be identical. The existing training strategy only pursues the metric of PQ retrieval, which destroys the representation of biencoder. As shown in Table 2, the results fall significantly in dense embedding retrieval. The multi-task loss optimizes both PQ retrieval performance and dense embedding performance, which can further aid in modeling discrete representations.

In Table 3, we use top documents retrieved by dense embeddings as ground truth and evaluate the performance of PQ retrieval on it, because we obtain quantized embeddings from dense ones approximately. By doing so, we can construct large test datasets and it is important for retrieval. Note that the three models in Table 3 are trained on the same training set belonging to MS MARCO Passage, but actually use three different test sets because they each have different recall results for dense embeddings retrieval. It shows that even when the answers are drawn from the models' own dense embeddings retrieval, our method still produces the best results. This suggests that our model does a good job of bringing the performance of PQ retrieval near to dense embeddings retrieval, which is the ability that other models do not have.

5.2.3 Impact of negative selection. The data distribution of sampled documents is plotted in Figure 3. At the end of the first phase

Table 1: Overall performances on MS MARCO Passage Retrieval and TREC2020 Deep Learning Track

Method	Ratio	MS MARCO Passage				DL Passage			
		MRR@10	MRR@100	Recall@10	Recall@30	NDCG@3	NDCG@5	NDCG@10	Recall@10
CoCodenser	1x	0.3815	0.3922	0.6650	0.8096	0.7461	0.7021	0.6803	0.2256
ScaNN	128x	0.0607	0.0679	0.1279	0.2136	0.1694	0.1683	0.1456	0.0473
PQ	128x	0.0582	0.0658	0.1275	0.2206	0.1792	0.1923	0.1835	0.0741
OPQ	128x	0.3030	0.3146	0.5676	0.7213	0.6333	0.6205	0.6042	0.2034
Distill-VQ	128x	0.3140	0.3250	0.5986	0.7471	0.6341	0.6264	0.5905	0.1902
JPQ	128x	0.3389	0.3503	0.6068	0.7592	0.7080	0.6812	0.6556	0.2218
RepCONC	128x	0.3497	0.3608	0.6155	0.7635	0.7081	0.6768	0.6541	0.2229
TMJ (ours)	128x	**0.3568**	**0.3671**	**0.6253**	**0.7650**	**0.7288**	**0.7069**	**0.6673**	**0.2275**

Table 2: Comparison between existing loss and multi-task loss on PQ retrieval and dense embeddings retrieval. The number of sub-vectors M is set to 24 and 48. PQ retrieval refers to using an index consisting of compressed embeddings to complete the search. Dense embedding retrieval refers to using an index consisting of uncompressed embeddings to complete the search.

Method	Sub	PQ Retrieval				Dense Embedding Retrieval			
		MRR@10	MRR@100	Recall@10	Recall@30	MRR@10	MRR@100	Recall@10	Recall@30
Conventional Loss of Joint Training	24	0.3428	0.3538	0.6137	0.7609	0.2632	0.2724	0.4771	0.5957
	48	0.3657	0.3766	0.6429	0.7887	0.2321	0.2396	0.4152	0.5127
Multi-task Loss	24	0.3514	0.3620	0.6190	0.7618	0.3769	0.3876	0.6530	0.7985
	48	0.3664	0.3771	0.6444	0.7906	0.3781	0.3889	0.6592	0.8025

Table 3: Evaluation of the quantized vectors under the condition that the candidates retrieved by the unquantized vectors are the answers. Top-K means using the first k candidates.

Method	Top-3		Top-5		Top-10	
	R@10	R@30	R@10	R@30	R@10	R@30
OPQ	0.7801	0.9156	0.7093	0.8827	0.5576	0.8048
RepCONC	0.5723	0.7061	0.5118	0.6602	0.3996	0.5722
TMJ	0.8779	0.9626	0.8195	0.9428	0.6589	0.8905

of training, the PQ retrieval achieves good performance on the test set, but it still underperforms than the dense retrieval from the same model. We define the top 200 documents retrieved by quantized embeddings as set S_c and the top 200 documents retrieved by uncompressed embeddings as set S_u. A document belonging to the S_d (or difference set) means that it can be covered by the result of PQ retrieval, but not by the result of dense embeddings retrieval. A document belonging to the S_i (or intersection) indicates that both embeddings can recall it. According to Figure 3, the ratio of intersection and difference set is about 127:71, and the top-ranked recall documents are more likely to belong to the intersection. As the ranking gets progressively larger, the possibility of documents belonging to the difference set gradually increases.

More experiments about sampling strategies is shown in Table 4. The selection of positive samples is important for further optimization of the model. The second row of the Table 4 shows the result based on the method of mining answer, i.e., using S_i as answers and S_d as negatives. This approach does not increase the performance, and to some extent illustrates the importance of ground truth. In the case of using ground truth documents, using the difference set as negatives is less effective than the other methods. Besides, using

Table 4: Impact of positives and negatives in the second stage. Mining Answer means using S_i as answers and S_d as negatives. Experiments in the bottom of the table use ground truth documents as answers. Top-200 means using top 200 documents retrieved by quantized embeddings as negatives. Same as top-200, Difference set and Intersection represent the source of negative samples respectively.

Method	MRR@10	MRR@100	R@10	R@100
Baseline	0.3514	0.3620	0.6190	0.8718
Mining Answer	0.3516	0.3623	0.6184	0.8738
GT+Top-200	0.3563	0.3668	0.6248	0.8762
GT+Difference set	0.3495	0.3603	0.6202	0.8773
GT+Intersection	0.3568	0.3671	0.6253	0.8780

the intersection as negative samples employs a smaller number of samples than using the top-200, but obtains the slightly better results. Such observations indicate that the top-ranked documents recalled by both PQ retrieval and dense embeddings retrieval are more likely to be hard negative samples.

6 EVALUATION IN INDUSTRIAL CONTEXTS

Given the difference in distribution between the private and public datasets, our setup of the experiments in this part differs from that in Section 5 in two ways, including the negative samples in the first stage and the addition of post-processing of the prediction results. Note that the overall training strategy remains the same. To avoid confusion, we will explain the differences in detail in this section.

Negative samples in stage 1. For the public benchmark, we adopt the common practice of previous works, using the ground truth given in the dataset and additionally mining negative samples.

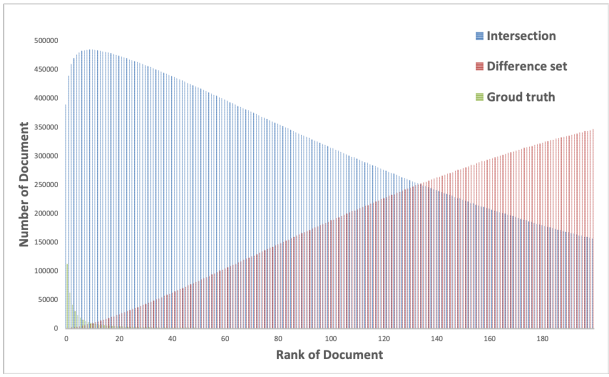

Figure 3: Distribution of negative sampling during training. The X-axis represents the rank of document when it was recalled, ranging from 1 to 200. The Y-axis indicates how many documents ranked K belong to the intersection/difference set/groud truth. The set of ground truth does not overlap with the other two sets.

Table 5: Retrieval top-K accuracy on the human-labeled industrial dataset. The dataset contains queries based on number of impressions.

Method	Low		Medium		High	
	R@5	R@10	R@5	R@10	R@5	R@10
OPQ	0.5868	0.7836	0.5899	0.7792	0.6452	0.8768
RepCONC	0.6198	0.8104	0.6020	0.8071	0.6712	0.9022
TMJ	0.6223	0.8208	0.6222	0.8111	0.6867	0.9159

In an industrial scenario, we annotate a large number of query-document pairs with relevance to make them useful for model training. These query-document pairs may include both low and high relevance pairs, so we directly use low-relevance documents as negative samples of the given query during the first stage. It should be noted that the negative samples used in the second stage on the private dataset are mined in the same way as on the benchmark. Both of them are mined using PQ-oriented Negative Mining.

Post-processing. It is crucial to maintain a high document recall rate for a web search system. However, the retrieval performance of most public PQ methods is not good enough. We improve our online search system by implementing a post-processing. Specifically, we will first recall a collection of candidates using an index constructed by compressed embeddings, and then rerank the collection using uncompressed embeddings. Taking Recall 2@20 as an example, the model first recalls the set of candidates of size 20, and takes only the top two results as the prediction results after reranking. The recall obtained by this method is better than the normal Recall@2. In this section, all experiments have the post-processing and the default candidate set size is 20.

6.1 Datasets

There are two private datasets, both of which are derived from the Baidu.com logs. One of the private datasets was manually annotated, with the annotators scoring the data on a correlation scale of 1 to 5. Another private dataset was generated by an automatic annotation method that used the top-K documents recalled by the uncompressed index as the real positives of queries.

Because the manually annotated dataset is derived from previous data, it is relatively outdated compared to the current online search data. And additional annotations would be required to expand it. The automatically annotated dataset is able to maintain distributional consistency with the online data at a lower cost. Specifically, we follow the steps below to create an automatically annotated dataset. First, we extract queries and candidate documents from the regularly updated databases. Second, we construct an uncompressed index consisting of dense vectors without quantization. Finally, we use the top-K documents retrieved by the uncompressed index as the real positives of queries. In addition to low annotation costs, the automatically annotated dataset has another advantage: it is possible to measure how much semantic representation the index loses during the compression process. Compared to uncompressed indexes, compressed indexes lose some effectiveness in exchange for efficiency improvements. The dataset uses the documents retrieved by the uncompressed index as ground truth. This helps to clearly identify this part of the loss.

Our proposed architecture significantly improves the recall rates in three scenarios, as shown in Table 5, which presents the performance of the different methods on the manually annotated dataset. The three test sets consist of different queries that are rarely searched (Low), often searched (Medium) and frequently searched (High) by users. Table 6 shows the performance of the different methods on the automatically annotated dataset. The previous supervised method performs worse than the unsupervised method, which can be explained by the fact that the conventional loss of joint training can be detrimental to the biencoder and thus reduce the effectiveness of the uncompressed vectors.

6.2 Efficiency

In practical web search scenarios, brute-force search is prohibitive due to its huge memory consumption and time cost. HNSW [30], the current SOTA for ANN search, is widely used by industrial search engines to overcome this problem. Due to the low coupling between the vector itself and the approximate nearest neighbor algorithm, we can easily combine HNSW with the Product Quantization.

To better simulate a real online scenario, we run an eight-minute stress test with a setting of 1000 queries per second. The results are shown in the table 7. The first row shows the performance of the retrieval system based on the HNSW algorithm. The second row shows the performance of the system with the addition of OPQ [10]. Since the main idea of PQ is to decompose the space into Cartesian products of low-dimensional subspaces and quantize each subspace separately, we can see reductions in retrieval time, CPU usage and index storage compared to the base version of HNSW, but it correspondingly loses some of the recall effectiveness. By applying learned PQ to an HNSW-based billion-scale information retrieval system, our scheme shows competitiveness in time and memory efficiency, as well as high-quality retrieval performance.

6.3 Online Evaluation

Furthermore, we compare the performance of TMJ with baseline under the same conditions by using human assessors to evaluate

Table 6: Rerank Top-2 accuracy on the auto-labeled samples. The sizes of the candidate sets are 2, 5, 10, 20.

Methods	2@2	2@5	2@10	2@20
OPQ	0.4655	0.6784	0.7827	0.8441
RepCONC	0.4071	0.6128	0.7343	0.8188
TMJ-Stage 1	0.5254	0.7453	0.8440	0.8935
TMJ-Stage 1&2	0.5761	0.8016	0.8991	0.9462

Table 7: Efficiency metrics during stress testing on the auto-labeled dataset. Time is the average retrieval time for each query. CPU usage is represented by relative values.

Methods	R@2	Time	CPU Usage	Memory
HNSW	**0.9538**	331μs	100%	3.54G
HNSW+OPQ	0.8441	299μs	94.25%	**1.34G**
HNSW+TMJ	0.9462	**279μs**	**87.35%**	**1.34G**

Table 8: Relative improvement on manual evaluation. LR-DCG denotes the DCG of documents with low relevance.

	Relative values		P-value	
	R@2	R@4	R@2	R@4
DCG	0.03%	0.02%	0.0700	0.0993
LR-DCG	-0.38%	-0.49%	0.1573	0.0339

the top-K ranking results in Baidu Web Search. The relative improvement validated by manual evaluation is reported in Table 8. Particularly, LR-DCG means the ratio of low relevance documents with 0 or 1 grades. In Table 8, the rise of DCG and the decline of LR-DCG mean that our proposed strategies do improve the effectiveness of retrieval.

7 CONCLUSION

This paper presents a Two-stage Multi-task Joint training technique (TMJ) to learn discrete document representations, which is simple and effective for real-world practical applications. An empirical exploration is made for the existing objective function of PQ, where we identify the limitation of using reconstruction loss and PQ retrieval loss. We propose a multi-task loss to regularize PQ centroid embeddings and preserve the retrieval quality of dense embeddings. Knowing that hard negatives are important for PQ learning, we introduce a novel negative sample mining method, which further utilizes the discrete document representations after PQ learning. The application on billion-scale web search and the experiment results on both public and private datasets validate the effectiveness of our proposed method.

REFERENCES

[1] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1798–1828.

[2] Yue Cao, Mingsheng Long, Jianmin Wang, and Shichen Liu. 2017. Deep visual-semantic quantization for efficient image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1328–1337.

[3] Ting Chen, Lala Li, and Yizhou Sun. 2020. Differentiable product quantization for end-to-end embedding compression. In *International Conference on Machine Learning*. PMLR, 1617–1626.

[4] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. arXiv:arXiv:2102.07662

[5] 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* (2018).

[6] Luyu Gao and Jamie Callan. 2021. Unsupervised Corpus Aware Language Model Pre-training for Dense Passage Retrieval. arXiv:arXiv:2108.05540

[7] Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. 2021. Complement lexical retrieval model with semantic residual embeddings. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part I 43*. Springer, 146–160.

[8] Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. 2021. Scaling deep contrastive learning batch size under memory limited setup. *arXiv preprint arXiv:2101.06983* (2021).

[9] Lianli Gao, Xiaosu Zhu, Jingkuan Song, Zhou Zhao, and Heng Tao Shen. 2019. Beyond product quantization: Deep progressive quantization for image retrieval. *arXiv preprint arXiv:1906.06698* (2019).

[10] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. 2013. Optimized product quantization. *IEEE transactions on pattern analysis and machine intelligence* 36, 4 (2013), 744–755.

[11] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. 1999. Similarity search in high dimensions via hashing. In *Vldb*, Vol. 99. 518–529.

[12] Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. 2022. Semantic models for the first-stage retrieval: A comprehensive review. *ACM Transactions on Information Systems (TOIS)* 40, 4 (2022), 1–42.

[13] Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. 2020. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning*. PMLR, 3887–3896.

[14] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*. PMLR, 3929–3938.

[15] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *KDD*. 2553–2561.

[16] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. In *International Conference on Learning Representations*.

[17] Suhas Jayaram Subramanya, Fnu Devvrit, Harsha Vardhan Simhadri, Ravishankar Krishnawamy, and Rohan Kadekodi. 2019. Diskann: Fast accurate billion-point nearest neighbor search on a single node. *Advances in Neural Information Processing Systems* 32 (2019).

[18] Herve Jegou, Matthijs Douze, and Cordelia Schmid. 2010. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence* 33, 1 (2010), 117–128.

[19] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

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

[21] Benjamin Klein and Lior Wolf. 2019. End-to-end supervised product quantization for image search and retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5041–5050.

[22] Sen Li, Fuyu Lv, Taiwei Jin, Guli Lin, Keping Yang, Xiaoyi Zeng, Xiao-Ming Wu, and Qianli Ma. 2021. Embedding-based product retrieval in taobao search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3181–3189.

[23] Wen Li, Ying Zhang, Yifang Sun, Wei Wang, Mingjie Li, Wenjie Zhang, and Xuemin Lin. 2019. Approximate nearest neighbor search on high dimensional data—experiments, analyses, and improvement. *IEEE Transactions on Knowledge and Data Engineering* 32, 8 (2019), 1475–1488.

[24] Yizhi Li, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2021. More robust dense retrieval with contrastive dual learning. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*. 287–296.

[25] Yiding Liu, Weixue Lu, Suqi Cheng, Daiting Shi, Shuaiqiang Wang, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained language model for web-scale retrieval in baidu search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3365–3375.

[26] 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* (2019).

[27] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

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

[29] Yury Malkov, Alexander Ponomarenko, Andrey Logvinov, and Vladimir Krylov. 2014. Approximate nearest neighbor algorithm based on navigable small world graphs. *Information Systems* 45 (2014), 61–68.

[30] Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence* 42, 4 (2018), 824–836.

[31] Priyanka Nigam, Yiwei Song, Vijai Mohan, Vihan Lakshman, Weitian Ding, Ankit Shingavi, Choon Hui Teo, Hao Gu, and Bing Yin. 2019. Semantic product search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2876–2885.

[32] Prafull Prakash, Julian Killingback, and Hamed Zamani. 2021. Learning Robust Dense Retrieval Models from Incomplete Relevance Labels. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1728–1732.

[33] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2020. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2010.08191* (2020).

[34] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223* (2019).

[35] Jinpeng Wang, Ziyun Zeng, Bin Chen, Tao Dai, and Shu-Tao Xia. 2022. Contrastive quantization with code memory for unsupervised image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 2468–2476.

[36] Shitao Xiao, Zheng Liu, Weihao Han, Jianjin Zhang, Defu Lian, Yeyun Gong, Qi Chen, Fan Yang, Hao Sun, Yingxia Shao, et al. 2022. Distill-VQ: Learning Retrieval Oriented Vector Quantization By Distilling Knowledge from Dense Embeddings. *arXiv preprint arXiv:2204.00185* (2022).

[37] Shitao Xiao, Zheng Liu, Yingxia Shao, Defu Lian, and Xing Xie. 2021. Matching-oriented product quantization for ad-hoc retrieval. *arXiv preprint arXiv:2104.07858* (2021).

[38] 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* (2020).

[39] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Jointly optimizing query encoder and product quantization to improve retrieval performance. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2487–2496.

[40] 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*. 1503–1512.

[41] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2022. Learning Discrete Representations via Constrained Clustering for Effective and Efficient Dense Retrieval. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1328–1336.

[42] Han Zhang, Hongwei Shen, Yiming Qiu, Yunjiang Jiang, Songlin Wang, Sulong Xu, Yun Xiao, Bo Long, and Wen-Yun Yang. 2021. Joint learning of deep retrieval model and product quantization based embedding index. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1718–1722.

[43] Han Zhang, Songlin Wang, Kang Zhang, Zhiling Tang, Yunjiang Jiang, Yun Xiao, Weipeng Yan, and Wen-Yun Yang. 2020. Towards personalized and semantic retrieval: An end-to-end solution for e-commerce search via embedding learning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2407–2416.

[44] Jianjin Zhang, Zheng Liu, Weihao Han, Shitao Xiao, Ruicheng Zheng, Yingxia Shao, Hao Sun, Hanqing Zhu, Premkumar Srinivasan, Weiwei Deng, et al. 2022. Uni-Retriever: Towards Learning The Unified Embedding Based Retriever in Bing Sponsored Search. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4493–4501.

[45] Yanzhao Zhang, Richong Zhang, Samuel Mensah, Xudong Liu, and Yongyi Mao. 2022. Unsupervised sentence representation via contrastive learning with mixing negatives. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 11730–11738.

[46] Kun Zhou, Yeyun Gong, Xiao Liu, Wayne Xin Zhao, Yelong Shen, Anlei Dong, Jingwen Lu, Rangan Majumder, Ji-Rong Wen, Nan Duan, et al. 2022. Simans: Simple ambiguous negatives sampling for dense text retrieval. *arXiv preprint arXiv:2210.11773* (2022).