Distill-VQ: Learning Retrieval Oriented Vector Quantization By Distilling Knowledge from Dense Embeddings

Shitao Xiao∗
stxiao@bupt.edu.cn
Beijing University of Posts and Telecommunications
Beijing, China

Zheng Liu
zhengliu@microsoft.com
Microsoft Research Asia
Beijing, China

Weihao Han
weihan@microsoft.com
Microsoft Search Technology Center
Beijing, China

Defu Lian
liandefu@ustc.edu.cn
University of Science and Technology of China
Hefei, China

Yeyun Gong
yegong@microsoft.com
Microsoft Research Asia
Beijing, China

Qi Chen
cheqi@microsoft.com
Microsoft Search Technology Center
Beijing, China

Hao Sun
hasun@microsoft.com
Microsoft Search Technology Center
Beijing, China

Yingxia Shao†
shaoyx@bupt.edu.cn
Beijing University of Posts and Telecommunications
Beijing, China

Xing Xie
xingx@microsoft.com
Microsoft Research Asia
Beijing, China

ABSTRACT
Vector quantization (VQ) based ANN indexes, such as Inverted File System (IVF) and Product Quantization (PQ), have been widely applied to embedding based document retrieval thanks to the competitive time and memory efficiency. Originally, VQ is 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 severe loss of retrieval quality. Recent works identify such a defect, and propose to minimize the retrieval loss through contrastive learning. However, these methods intensively rely on queries with ground-truth documents, whose performance is limited by the insufficiency of labeled data.

In this paper, we propose Distill-VQ, which unifies the learning of IVF and PQ within a knowledge distillation framework. In Distill-VQ, the dense embeddings are leveraged as “teachers”, which predict the query’s relevance to the sampled documents. The VQ modules are treated as the “students”, which are learned to reproduce the predicted relevance, such that the reconstructed embeddings may fully preserve the retrieval result of the dense embeddings. By doing so, Distill-VQ is able to derive substantial training signals from the massive unlabeled data, which significantly contributes to the retrieval quality. We perform comprehensive explorations for the optimal conduct of knowledge distillation, which may provide useful insights for the learning of VQ based ANN index. We also experimentally show that the labeled data is no longer a necessity for high-quality vector quantization, which indicates Distill-VQ’s strong applicability in practice. The evaluations are performed on MS MARCO and Natural Questions benchmarks, where Distill-VQ notably outperforms the SOTA VQ methods in Recall and MRR. Our code is available at https://github.com/staoxiao/LibVQ.

CCS CONCEPTS
• Computing methodologies → Continuous space search.

KEYWORDS
Vector Quantization, Knowledge Distillation, Embedding Based Retrieval, Approximate Nearest Neighbour Search

Permission to make digital or hard copies of all or part of the work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGIR ’22, July 11–15, 2022, Madrid, Spain
© 2022 Copyright held by the owner/authors. Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8732-3/22/07... $15.00
https://doi.org/10.1145/3477495.3531799

1513
1 INTRODUCTION

The embedding based retrieval (EBR) of documents plays an important role in many web applications, such as search engines and recommender systems [5, 12, 17, 23]; given an input query, the relevant documents are selected from the entire corpus based on embedding similarity. Knowing that brute-force linear scan is temporally infeasible, the embeddings need to be organized by ANN index in order to support real-world information retrieval. For the past decade, the vector quantization (VQ) techniques, e.g., Inverted File System (IVF) [1, 2, 19] and Product Quantization (PQ) [18, 40, 45], are widely applied, which enables ANN to be performed with competitive time and memory efficiency.

- Typical learning paradigms of VQ. VQ is typically learned to minimize the reconstruction loss, i.e., the minimization of 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. In recent works [45, 50], the dense embeddings and VQ are jointly learned to minimize the post-quantization retrieval loss. However, the existing methods typically rely on contrastive learning, which calls for a tremendous amount of queries labeled with ground-truth documents. As a result, it may suffer from inferior performance due to the insufficiency of labeled data.

- Our Solution. In this paper, we propose Distill-VQ, which jointly learns IVF and PQ via knowledge distillation. In Distill-VQ, we leverage well-trained dense embeddings\(^1\) as the teachers: for each query, the teachers will predict its relevance scores towards a set of candidate documents. The predicted scores are used as the indicators of how likely the corresponding documents are the ground-truth to the query. The quantization modules are treated as the students, which are learned to reproduce the teachers’ predictions, such that the dense embeddings’ retrieval performance can be effectively preserved. Compare with the existing methods, Distill-VQ gives rise to the following merits.

First and foremost, Distill-VQ is able to exploit the data more effectively, which will significantly contribute to the retrieval quality. The current joint learning methods are typically based on contrastive learning and driven by labeled data: the model is learned to identify the document labeled as ground-truth to each query, where the remaining documents within the corpus are equally treated as negative samples. In contrast, Distill-VQ is to learn from the teachers (i.e., the well-trained dense embeddings), thus relaxing the limitation from labeled data. During the knowledge distillation process, the high-relevance documents to each query, whether labeled as ground-truth or not, will probably be given high relevance scores from the teachers. As a result, such documents will virtually serve as positive samples, which are huge supplements to the original labeled dataset. Besides, instead of treating all remaining documents equally as negative samples, Distill-VQ will penalize more for the documents which are predicted to be less relevant with the query. Therefore, it may also alleviate the false negative problems existed in the conventional methods.

In addition, Distill-VQ also contributes to higher applicability while working with those off-the-shelf document embeddings, e.g., CDSSM [24] and SBERT [35]. As discussed, the existing joint learning methods typically learn dense embeddings and VQ at the same time leveraging massive-scale of labeled data. However, it remains little explored of how to optimize the VQ’s performance on top of well-trained dense embeddings. Distill-VQ naturally alleviates this problem as it is designed to work with fixed document embeddings. Besides, knowing that the model is learned from knowledge distillation, the labeled data is no longer a necessity. In fact, it is experimentally verified that Distill-VQ may generate high-quality vector quantization without using ground-truth documents, which significantly relaxes the requirement on training data.

We perform comprehensive explorations for the optimal conduct of knowledge distillation, including the document sampling strategies (i.e., which documents to distill the teachers’ knowledge), and the function to preserve teachers and students’ similarity (which defines how to distill teachers’ knowledge from their predictions to the sampled documents). It is found that distilling knowledge from a sufficient large number of Top-K documents (the Top-K relevant documents to each query predicted by the teachers) with functions enforcing ranking order invariance, e.g., ListNet [4], may produce the most effective training outcome. Our evaluations are performed on two large-scale benchmarks: MS MARCO and Natural Questions, where Distill-VQ outperforms the SOTA vector quantization methods by notable margins. We also integrate Distill-VQ’s output into FAISS [22], which significantly improves the performance of its original VQ-based indexes, e.g., IVFPQ, IVF-Flat; and enables our method to be directly applicable to real-world applications. To summarize, our work is highlighted with the following points.

- We propose a novel framework Distill-VQ, which jointly learns IVF and PQ for the optimization of their retrieval performance.

\(^1\)We target on optimizing the retrieval performance resulted from vector quantization. How to train dense embeddings is beyond the discussion of this paper.
By distilling knowledge from the well-trained dense embeddings, Distill-VQ achieves much more effective exploitation of the data, and contributes to a higher applicability when working with off-the-shelf embeddings.

Comprehensive explorations are performed for the optimal conduct of Distill-VQ, including the document sampling strategies and the functions to enforce teachers and students’ consistency.

Experimental studies on MS MARCO and Natural Questions benchmarks verify the effectiveness of Distill-VQ.

2 RELATED WORKS

Embedding based retrieval has been widely applied in practice, such as search engines [30, 39], question answering [33, 34, 54], online advertising [26, 31], and content-based recommender systems [27, 44]. Knowing that the documents need to be retrieved from a large-scale corpus, where brute-force linear scan will be temporally infeasible, it calls for approximate nearest neighbour search (ANN) [28] such that documents with high embedding similarities can be efficiently selected. The ANN index needs to trade-off three key factors: time cost, memory usage, and recall rate. That’s to say, it is expected to recall high-quality candidates with the minimum consumption of time and memory, and recall rate. The workflow of Distill-VQ is divided into four steps. Firstly, Distill-VQ achieves much more effective exploitation of the embeddings for all documents in the offline stage. The framework of Distill-VQ is shown as Figure 2, where both IVF and PQ modules are optimized. As a result, the developed techniques can be directly applied to generic VQ-based ANN indexes, and PQ modules are optimized. It is generally believed that KD contributes to the representation learning thanks to the exploitation of massive unlabeled data [46] and label smoothing [48]. Distill-VQ inherits such merits of KD, thus becoming more effective in preserving the dense embeddings’ retrieval performance.

3 DISTILL-VQ

The framework of Distill-VQ is shown as Figure 2, where both IVF and PQ modules are optimized. As a result, the developed techniques can be directly applied to generic VQ-based ANN indexes, such as PQ, IVFPQ, IVF-Flat, IVFADC-G-P, DiskANN, SPANN, etc. The workflow of Distill-VQ is divided into four steps. Firstly, Distill-VQ works with a well-trained document encoder, which generates the embeddings for all documents in the offline stage. The document embeddings are fixed in Distill-VQ, which longer take any computation cost in quantization learning. Second, each document embedding is assigned to its nearest centroid in IVF and the nearest codeword in each codebook of PQ. The assignment results are represented in binarized forms (i.e., one-hot vectors indicating the assigned ID), and used to generate the reconstructed embeddings for the document. Thirdly, the query encoder is used to generate query embedding, which will interact (inner-product) with the reconstructed document embedding for the relevance score. Finally, the knowledge distillation is performed, which enables the relevance scores computed by students (i.e., the reconstructed embeddings) to imitate the predictions from the teachers (i.e., the well-trained dense embeddings of the queries and documents).

3.1 Quantization and Reconstruction

The document embeddings are quantized and reconstructed via the following steps. Firstly, the document embedding \( d \) is generated for each document and the embedding models via contrastive learning, in which the discrimination of the ground-truth documents for each query can be optimized (referred as retrieval loss minimization), as Figure 1 (C). Compared with the previous minimization of the reconstruction loss, the retrieval loss minimization achieves substantial improvements in terms of the retrieval performance.

**Distinctions of our work.** Distill-VQ is similar with the existing works as it also targets on optimizing the retrieval quality rather than simply minimizing the reconstruction loss. However, Distill-VQ is differentiated in the following aspects, as Figure 1 (D). First and foremost, Distill-VQ is learned from well-trained dense embeddings via knowledge distillation, while the existing methods rely on contrastive learning. Secondly, Distill-VQ relaxes the requirement on labeled data, and can be effectively trained purely with unlabeled data. Thirdly, Distill-VQ works with fixed document embeddings instead of end-to-end optimization of the whole model.

**Knowledge Distillation.** Finally, it should be noted that knowledge distillation (KD) [14] plays the fundamental role in Distill-VQ. KD used to be a powerful tool for model compression [21, 38, 41], where the student model (lightweight) is learned to imitate the predictions from the teacher model (heavyweight). In recent years, it becomes increasingly popular to apply KD for representation learning. For example, in [16, 36, 52], the representation models are distilled from the relevance scores predicted by the re-ranking models; and in [7, 46], the representation models are pretrained by distilling knowledge from the pseudo labels annotated by the teacher models. It is generally believed that KD contributes to the representation learning thanks to the exploitation of massive unlabeled data [46] and label smoothing [48]. Distill-VQ inherits such merits of KD, thus becoming more effective in preserving the dense embeddings’ retrieval performance.
3.2 Knowledge Distillation

The reconstructed embeddings are learned from knowledge distillation, where the well-trained embeddings for the document ($v_d$) and query ($v_q$) are utilized as the Teachers. Particularly, for each query ($q$), the teachers’ scores ($T_q$) are computed for its sampled candidate documents ($d \in D_q$) based on $v_q$ and $v_d$:

$$T_q \leftarrow \{ (v_q, v_d) | d \in D_q \}.$$  

where “($\cdot$)” is the inner-product operator. The reconstructed embeddings are treated as the Students, which will generate their scores ($S_q^\alpha$ and $S_q^\beta$) towards each query’s candidate documents:

$$\text{IVF}: S_q^\alpha \leftarrow \{ (\tilde{v}_q, \tilde{v}_d) | d \in D_q \};$$

$$\text{PQ}: S_q^\beta \leftarrow \{ (v_q, \tilde{u}_d + \hat{v}_d) | d \in D_q \}.$$  

(5)

Note that the query encoder is still learnable in knowledge distillation. Therefore, the query embedding $v_q$ will become different from its original value $v_q$ (i.e., the one used for computing teachers’ scores). The students’ scores are required to preserve the teachers’ predictions w.r.t. a certain similarity function $\langle \cdot, \cdot \rangle$. Finally, the objective for the knowledge distillation is formulated as:

$$\min_{O, C, S_q^\alpha, S_q^\beta} - \sum_q \left\{ f(T_q, S_q^\alpha | D_q) + f(T_q, S_q^\beta | D_q) \right\}.$$  

(6)

The above optimization problem involves two groups of learnable parameters: the IVFPQ related parameters $O$ and $C$, and the query encoder (which generates $\tilde{v}_q$). Compared with the conventional joint learning methods [45, 50], the document embedding $v_d$ is fixed in Distill-VQ, meaning that the document encoder is excluded from the training process. Considering that the document is usually much larger than the query in terms of sequence length, and there are multiple candidate documents sampled for each query, such a difference will lead to a major saving of the computation cost.

Given the above formulation, the vector quantization is learned through the following iterative process (Alg. 1): for each query, the candidate documents $D_q$ are sampled from the entire documents $D$; then, teachers’ scores are computed based on the well-trained dense embeddings, and the students’ scores are computed based on the reconstructed embeddings; finally, the model is updated for the minimization of the distillation loss through gradient descent.

The knowledge distillation in Alg. 1 involves two critical factors: 1) the Similarity Function $f(\cdot, \cdot)$, which determines what kind of similarity to be preserved between the teachers and students; and 2) the Sampling Strategy for the candidate documents $D_q$, which identifies the documents where the similarity needs to be preserved.
In the next part, we’ll make discussions on a series of alternative solutions for both factors, whose impacts will be comprehensively explored in our experimental studies.

### 3.2.1 Similarity Function

The students may preserve the teachers' predictions in two alternative ways. One is to enforce the Score Invariance, where the students are expected to generate the same relevance scores for the candidate documents as predicted by the teachers. It is a quite rigid requirement for the similarity between teachers and students. In this place, we’ll introduce the following similarity functions for the enforcement of score invariance.

- **Mean Square Error (MSE).** The simplest form of score invariance is to minimize the absolute difference between the scores predicted by the teachers and students:

\[
f(T_q, S_q^d|D_q) = \sum_{d \in D_q} ||s_{q,d} - t_{q,d}||,
\]

where \(S_q^d\) are the students' predicted scores, i.e., \(S_q^t\) or \(S_q^s\); \(s_{q,d}\) and \(t_{q,d}\) are the students' and teachers' predictions towards each candidate document \(d\), as Eq. 4 and 5.

- **Margin MSE.** A modification of MSE is utilized in [16]: instead of directly minimizing the absolute difference, it minimizes the difference between the margins of scores for two candidate documents \(d_1\) and \(d_2\), as predicted by the teachers and students:

\[
f(T_q, S_q^d|D_q) = \sum_{d_1, d_2 \in D_q} ||(t_{q,d_1} - t_{q,d_2}) - (s_{q,d_1} - s_{q,d_2})||.
\]

Margin MSE is a relaxation of MSE, which emphasizes more of the pairwise relationships between the candidate documents rather than the absolute scores.

Aside from the above functions for the enforcement of score invariance, a more flexible option is to enforce the Ranking Order Invariance, where the students are merely expected to generate the same ranking orders for the candidate documents as predicted by the teachers. It is a highly relaxed requirement for the teachers and students similarity, which will probably become much easier to optimize. In this place, we introduce the following similarity functions to enforce the ranking order invariance.

- **RankNet.** The weighted RankNet (RankNet for short) was a pairwise ranking loss proposed in [15]. With RankNet, the teachers and students' consistency is to be kept in terms of the pairwise relationship between two candidate documents:

\[
f(T_q, S_q^d|D_q) = - \sum_{d_1, d_2 \in D_q} \left( t_{q,d_1} - t_{q,d_2} \right) \ast \log \sigma(s_{q,d_1} - s_{q,d_2}).
\]

It can be regarded as a variation of BPR loss [37] weighted by the teachers' prediction. Aside from the pairwise ranking loss, the functions on listwise similarity may also be leveraged.

- **KL-divergence.** The teachers and students’ predicted scores towards the candidate documents are normalized into the following distributions (teachers: \(\Phi_q^t\), students: \(\Phi_q^s\)):

\[
\phi_q^t = \frac{\exp(t_{q,d})}{\sum_{d \in D_q} \exp(t_{q,d})}, \quad \phi_q^s = \frac{\exp(s_{q,d})}{\sum_{d \in D_q} \exp(s_{q,d})}.
\]

A sufficient condition for ranking order invariance is that the students may generate the same distribution as the teachers. To this end, the KL-divergence is minimized for the above distributions:

\[
f(T_q, S_q^d|D_q) = \text{KL}(\Phi_q^s||\Phi_q^t) = - \sum_{d \in D_q} \phi_q^{t,d} \log \frac{\phi_q^{t,d}}{\phi_q^{s,d}}.
\]

- **ListNet.** A close variant of KL-divergence is ListNet [4], which is a well-known function for ranking order preservation. In ListNet, the cross-entropy is minimized for the students and teachers' distribution over the candidate documents (as Eq. 10):

\[
f(T_q, S_q^d|D_q) = H(\Phi_q^s, \Phi_q^t) = - \sum_{d \in D_q} \phi_q^{t,d} \log \phi_q^{s,d}.
\]

Both KL-divergence and ListNet require the listwise similarity between teachers and students. What’s more, the teachers and students’ similarities on the top-ranked documents can be naturally emphasized by both functions, which enables knowledge distillation to be more consistent with the retrieval metrics.

### 3.2.2 Candidate Document Sampling

The retrieval performance (reflected by metrics like Recall@K and MRR) is mainly determined by the top-ranked documents, i.e., whether the ground-truth can be covered by the Top-K retrieved documents. Therefore, the students are expected to emphasize more of its similarity with the teachers on such groups of documents. Intuitively, the following sources of candidate documents can be applied for this purpose.

- **Ground-Truth.** One straightforward option is to utilize the ground-truth documents of each query, as these documents are the targets to be retrieved. Ideally, the teachers’ retrieval performance can be fully preserved if the students may always generate the same rankings for the ground-truth documents (as the teachers). However, it will probably take a huge amount of labeled data to ensure the knowledge distillation quality, which will be a challenging condition in reality.

- **Top-K Documents.** Another other option is to select the documents with the Top-K similarities to the query as the candidates (based on the original dense embeddings):

\[
D_q = \text{Top-K}\{|v_q, v_d|d \in D\}.
\]

The Top-K documents are plausible for two reasons. Firstly, the students will fully preserve the teachers’ retrieval performance as long as they may generate the same Top-K documents. Secondly, it relaxes the limitation from labeled data, since plenty of training samples can be generated for knowledge distillation by getting the Top-K documents of arbitrary queries.

- **In-Batch Documents.** Aside from the top-ranked documents, the lower-ranked documents need to be introduced for the contrastive purpose, i.e., to make sure the lower-ranked documents
from the teachers are safely excluded from the students retrieval results. In this place, we leverage in-batch sampling [23], where one query will make use of other queries’ related documents within the same batch for knowledge distillation.

Finally, we may come up with the following sampling strategies.

- **All-Mixed.** The first one still uses labeled data, where the candidate documents are a mixture of all three sources: 1) ground-truth documents, 2) Top-K documents, 3) in-batch documents.
- **Top-K and In-Batch.** The second one purely works with the unlabeled data, where the candidate documents are merely a mixture of the Top-K and in-batch documents.

The impacts of both strategies are extensively analyzed in our experiment, together with explorations of other influential factors, such as the value of $K$, and how many documents to sample from the Top-K, etc. It is interesting to find that purely working with the unlabeled data, i.e., the Top-K and In-Batch combination, is sufficient to achieve highly competitive knowledge distillation performance.

### 4 EXPERIMENTS

#### 4.1 Experiment Settings

##### 4.1.1 Datasets.** Our experimental studies are based on the two popular benchmarks on document retrieval. The first one is the **MS MARCO (passage retrieval)** [32]. It is a widely used benchmark on web search and embedding-based retrieval, where the targeted answers from MS MARCO corpus need to be retrieved for queries from Bing search. The second one is the **Natural Questions (NQ)** [25]. The queries are real-world questions collected from Google Search; each of the queries is paired with an answer span and ground-truth passages from the Wikipedia pages. The detailed specifications of the datasets are shown as Table 1. For both datasets, we expect the ground-truth answer to each query can be retrieved from the entire corpus. Therefore, the experiment performances will be measured by the recall rate for the top-K retrieval result. Besides, we’ll also report MRR@10 for more comprehensive evaluation.

#### 4.1.2 Baseline methods.** We consider the following groups of baselines in experiments (corresponding to the experiments in Section 2). Firstly, the typical vector quantization methods: **IVFPOQ** [19], **IVFOPQ** [11], and **ScaNN** [13], where the vector quantization is learned for reconstruction loss minimization with fixed document embeddings. Secondly, the latest work on the joint learning of embedding models and vector quantization for reconstruction loss minimization: **Poem** [53]. Thirdly, the latest work on the joint learning of embedding models and vector quantization for retrieval loss minimization: **JPQ** [50] and **RepCONC** [51]. Note that JPQ and RepCONC utilize the original IVF provided by FAISS rather than learn it as the PQ component. We further introduce a variant of **MoPQ**[45] for ablation study, which optimizes the IVF and PQ both by contrastive learning.

##### 4.1.3 Implementation details.** Distill-VQ (our approach) leverages BERT-based backbone [9] for text encoding. In our experiments, we work with the released well-trained encoders from the latest works: **AR2-G** [52] and **CoCondenser** [10], which are the most accurate models on MS MARCO and NQ benchmarks. For the sake of fair comparisons, all the baseline methods will use the same **document embeddings** as Distill-VQ (which will be more accurate than their original implementations). For PQ, OPQ, ScaNN: the quantization will be directly applied to document embeddings from AR2-G and CoCondenser; for Poeem: the query encoder and the quantization module will be jointly learned for reconstruction loss minimization; for JPQ, RepCONC, MoPQ: the query encoder and the quantization module will be jointly learned for retrieval loss minimization. We wrap up the learned quantization results with the IVFPQ index implemented by **FAISS** [22], which enables document retrieval to be efficiently performed. The default configuration of index is specified as follows. For IVF: a total of 10,000 posting lists are deployed, of which the Top-100 posting lists (i.e., the Top 1%) will be visited for ANN search; for PQ, a total of 64 codebooks are deployed, each of which uses 8-bit codewords (i.e., 256 codewords per codebook). Aside from the default configuration, which we will also explore more settings of IVF and PQ in our experiments.

The experiments are based on clusters of 8” NVIDIA-A100-40GB GPUs and 2* AMD EPYC 7V12 64-Core CPUs. The algorithms are implemented with python 3.6 and PyTorch 1.8.0. We optimize the parameters with the AdamW optimizer. The learning rates for query encoder, IVF centroids and PQ codebooks are 5e-6, 1e-3 and 1e-4 respectively. More training details can be found in our released code: https://github.com/staoxiao/LibVQ.

#### 4.1.4 Problems to be explored.** Our experiments focus on three problems. First and foremost, Distill-VQ’s impact on retrieval quality against the existing VQ methods (Section 4.2.1). Secondly, analysis of knowledge distillation: the impact from different similarity functions and different sampling strategies for candidate documents (Section 4.2.2). Thirdly, exploration of Distill-VQ’s impact on each individual component: IVF and PQ, respectively, with variant configurations of both components (Section 4.2.3, 4.2.4).

#### 4.2 Experiment Analysis

##### 4.2.1 Overall performance.** The overall performances of different methods are evaluated under the default settings (IVF: search Top-100 out of 10,000 posting lists; PQ: 64 codebooks, each codebook is 8-bit). All the methods utilize the same document embeddings from AR2-G and CoCondenser for fair comparisons. Distill-VQ’s general effectiveness can be verified given its huge advantages over the baselines (as Table 2). For MS MARCO: Distill-VQ achieves 0.3607 and 0.3523 on MRR@10 with AR2-G and CoCondenser, outperforming the strongest baselines by +1.36% and +1.53%, respectively. For NQ: Distill-VQ achieves 0.6235 and 0.5781 on MRR@10 with both embeddings, leading to +4.0% and +0.92% over the strongest baselines. The above performances are also the SOTA results of their kinds as compared with the reported results in the latest works [43, 50, 51]. Note that our evaluation is performed on top of IVF.

| Datasets                  | MS MARCO Passage | Natural Questions |
|---------------------------|------------------|-------------------|
| #Train (original)         | 502,939          | 79,168            |
| #Train (processed)        | 502,939          | 58,880            |
| #Dev                      | 6,980            | 8,757             |
| #Test                     | –                | 3,610             |
| #Documents                | 8,841,823        | 21,015,324        |

Table 1: Specifications of the datasets.
with merely 1% of the posting lists searched, which means there is a 100x acceleration over the previous works.

Besides, it is also interesting to find that although the joint learning baselines (Poem, JPQ, RepCONC, MoPQ) are still generally better than PQ, OPQ and ScaNN, they are not as significant as the observations in previous works [43, 50, 51]. Meanwhile, the performances of Poem (minimizing the reconstruction loss) and other joint learning methods (minimizing the retrieval loss) are also close to each other in many cases. Much of these observations is probably due to the enhancement of document embeddings, as AR2-G and CoCondenser are more accurate than those enforcing the score invariance (MSE, Margin-MSE). However, the joint learning methods for retrieval loss minimization may also be applied to explain Margin-MSE’s advantage over MSE. We may be observed that different similarities do lead to distinct distillation effects, which can be distilled more easily. The same reasons may also be applied to explain Margin-MSE’s advantage over MSE. We may observe that ListNet remains competitive on both of the datasets; in contrast, KL-Divergence and ListNet are compared against each other. It can be observed that different similarities do lead to distinct distillation effects, where the overall performances from enforcing the ranking order invariance (KL-Div, ListNet) are comparatively better than those enforcing the score invariance (MSE, Margin-MSE). As discussed, the ranking order invariance is a sufficient condition for preserving dense embeddings’ retrieval performance. Besides, it is a largely relaxed requirement compared with the score invariance, which can be distilled more easily. The same reasons may also be applied to explain Margin-MSE’s advantage over MSE. We may observe that ListNet remains competitive on both of the datasets; in contrast, KL-Div and RankNet only achieve competitive

### 4.2.2 Explorations of knowledge distillation

We further explore the similarity function and candidate documents' impact on knowledge distillation. The remaining experiments will be based on AR2-G unless otherwise specified, given the similar observations between the two embeddings.

- **Similarity function’s impact** The evaluation results are shown in the upper part of Table 3, where the five alternative functions discussed in Section 3.2.1: MSE, Margin-MSE, RankNet, KL-divergence, and ListNet, are compared against each other. It can be observed that different similarities do lead to distinct distillation effects, where the overall performances from enforcing the ranking order invariance (KL-Div, ListNet, RankNet) are comparatively better than those enforcing the score invariance (MSE, Margin-MSE). As discussed, the ranking order invariance is a sufficient condition for preserving dense embeddings’ retrieval performance. Besides, it is a largely relaxed requirement compared with the score invariance, which can be distilled more easily. The same reasons may also be applied to explain Margin-MSE’s advantage over MSE. We may observe that ListNet remains competitive on both of the datasets; in contrast, KL-Div and RankNet only achieve competitive

### Table 2: Overall Performances on MS MARCO Passage Retrieval and Natural Questions, with embeddings from AR2-G and CoCondenser.

| Encoder       | Method       | MRR@10 | R@10  | R@50  | R@100 | MRR@10 | R@10  | R@50  | R@100 |
|---------------|--------------|--------|-------|-------|-------|--------|-------|-------|-------|
| AR2-G [52]    | IVFPQ        | 0.2403 | 0.4545| 0.6615| 0.7325| 0.2823 | 0.5002| 0.6764| 0.7240|
|               | IVFOPQ       | 0.3488 | 0.6003| 0.7881| 0.8385| 0.5505 | 0.7268| 0.8047| 0.8279|
|               | ScaNN        | 0.2670 | 0.4880| 0.6953| 0.7963| 0.4137 | 0.6470| 0.7742| 0.8124|
|               | Poem         | 0.3448 | 0.6077| 0.7975| 0.8487| 0.5588 | 0.7443| 0.8246| 0.8445|
|               | JPQ          | 0.3451 | 0.6107| 0.8029| 0.8558| 0.5666 | 0.7512| 0.8278| 0.8504|
|               | RepCONC      | 0.3449 | 0.6106| 0.7983| 0.8497| 0.5500 | 0.7382| 0.8141| 0.8373|
|               | MoPQ         | 0.3471 | 0.6119| 0.8037| 0.8564| 0.5835 | 0.7634| 0.8412| 0.8606|
|               | Distill-VQ   | **0.3607** | **0.6276** | **0.8221** | **0.8719** | **0.6235** | **0.7927** | **0.8627** | **0.8783** |
| CoCondenser [10] | IVFPQ       | 0.2252 | 0.4380| 0.6597| 0.7202| 0.5871 | 0.6177| 0.7642| 0.8047|
|               | IVFOPQ       | 0.3340 | 0.5947| 0.7810| 0.8325| 0.5454 | 0.7362| 0.8385| 0.8626|
|               | ScaNN        | 0.2470 | 0.4662| 0.6715| 0.7442| 0.4520 | 0.6540| 0.7831| 0.8193|
|               | Poem         | 0.3363 | 0.6026| 0.7908| 0.8403| 0.5652 | 0.7476| 0.8354| 0.8573|
|               | JPQ          | 0.3360 | 0.5954| 0.7810| 0.8335| 0.5436 | 0.7448| 0.8335| 0.8551|
|               | RepCONC      | 0.3319 | 0.5895| 0.7798| 0.8342| 0.5544 | 0.7495| 0.8365| 0.8602|
|               | MoPQ         | 0.3370 | 0.6056| 0.8019| 0.8542| 0.5689 | 0.7549| 0.8395| 0.8627|
|               | Distill-VQ   | **0.3523** | **0.6212** | **0.8086** | **0.8612** | **0.6235** | **0.7927** | **0.8627** | **0.8783** |

Table 3: Knowledge distillation analysis. Upper: impact from similarity functions. Lower: impact from sampled documents (GT: ground-truth; IB: in-batch sampling; Top-K1/Top-K2: sampling 1/2 document from Top-K; Top-K: using all Top-K documents; (*): w.o. ground-truth.)
with *). We may observe that the combination of in-batch sampling and the lower-ranked documents from in-batch sampling are still necessary for knowledge distillation. Comparing Top-3 1 against IB+Top-100, we may have the following findings from the demonstrated results.

Firstly, the labeled data is no longer a necessity in knowledge distillation, and highly competitive retrieval performance can be achieved by distilling knowledge from the unlabeled data (marked with *). We may observe that the combination of in-batch sampling (IB) and Top-K documents outperforms the other sampling strategies in comparison, especially those simply working with ground-truth and in-batch. Besides, it is also interesting to find that the retrieval quality tends to benefit more from using an increased amount of Top-K documents: using the entire Top-200 is much more effective than simply using a few samples from them (IB+Top-200 against IB+Top-200^2), and using Top-100 and Top-200 outperform the one simply using Top-10 (IB+Top-100/200 against IB+Top-10).

Secondly, although the top-ranked documents are important, the lower-ranked documents from in-batch sampling are still necessary for knowledge distillation. Comparing Top-3 1 + Top-200^1 1 with IB+Top-3 1 + Top-200^1 1 , the retrieval quality drops a lot on both datasets when in-batch sampling is disabled. However, it is also undesirable to simply work with the in-batch documents due to the lack of top-ranked documents, as the performance from IB alone is much lower than other methods.

4.2.3 Distill-VQ’s impact on IVF. Detailed exploration of Distill-VQ’s impact on IVF is shown in Table 4 and 5. To focus on the impact solely from IVF, all methods in comparison utilize the same product quantization: OPQ, as implemented by FAISS. A total of four methods are compared: 1) IVF: the original IVF implemented by FAISS, 2) IVF (Rctr): the IVF learned for reconstruction loss minimization (Poeem), 3) IVF (Rtrv): the IVF learned for retrieval loss minimization (MoPQ, and 4) IVF (Distill): the IVF learned from knowledge distillation (Distill-VQ).

For evaluations in Table 4, a total of 10,000 posting lists are deployed in IVF. The Top-K posting lists are searched for document retrieval, where a smaller K indicates a larger acceleration ratio, e.g., Top-10 indicates 1,000× speedup up (10,000/10). According to the experiment results, the retrieval quality can be continuously increased for all the methods with the growth of K; during this process, IVF (Distill) maintains notable advantages over other baselines. Besides, IVF (Distill) achieves more significant improvements when larger acceleration ratios are adopted (i.e., smaller K): when K equals to 1, IVF (Distill) outperforms the baselines by +2~6% on MS MARCO, and +8~17% on NQ. The advantages gradually diminish to +1% on both datasets when K increases to 500.

It is also interesting to see that both IVF (Rtrv) and IVF (Rctr) are no better than the original IVF when a large number of posting lists can be searched (e.g., K=500). This is probably because the jointly learning methods target on optimizing the Top-1 posting list of
Table 6: Distill-VQ’s Impact on PQ, with the size of codebooks expanded from $2^{x8}$ to $32^{x8}$ (#Codebooks × Bit_Per_Codebook). All methods in comparison use the same IVF generated by FAISS (search Top-100 out of 10,000 posting lists for 100× acceleration).

Figure 3: Efficiency and Retrieval Quality. I-IV: Recall@100 at different time/memory efficiency by changing the size of codebooks. V-VIII: Recall@100 at different time efficiency by changing 1) #Search: search the Top-#Search out of the 10,000 deployed posting lists, and 2) #Posting: the number of deployed posting lists (with the Top-1% searched).

each query, whereas caring little of the retrieval quality from the Top-K posting lists.

Secondly, the number of deployed posting lists is increased from 100 to 100,000 in Table 5. The document retrieval will always be made by searching the top 1% posting lists (e.g., the Top-1 will be searched for 100, and the Top-50 will be searched for 5,000), thus keeping a fixed acceleration ratio of 100×. For both MS MARCO and NQ, IVF (Distill) achieves notable improvements over the strongest baselines in every setting. Besides, it is interesting to observe that although the acceleration ratio is fixed, the performances are gradually improved when more posting lists are deployed. Since an observation is consistent with the previous finding [1, 2] that fine-grained inverted indexes are beneficial to retrieval quality. The search for the Top-K indexes can be expensive when a large number of posting lists are deployed; in [2], the problem is mitigated by establishing another ANN index (e.g., HNSW) for the posting lists.

4.2.4 Distill-VQ’s impact on PQ. Further exploration of Distill-VQ’s impact on PQ is made as Table 6. To focus on the impact solely from PQ, all methods in comparison utilize the same IVF generated by FAISS. The number of the codebooks is increased from 2 to 32: a smaller scale of codebooks means a larger compression ratio. With the growth of codebooks scale, the retrieval quality can be consistently improved for all the methods. When small-scale codebooks are used (meaning that the compression ratio is large), Distill-VQ is able to outperform all the baselines with notable advantages in both Recall and MRR. Besides, we may also compare different baselines more clearly when small-scale codebooks are used (meaning that the compression ratio is large), Distill-VQ is able to outperform all the baselines with notable advantages in both Recall and MRR. Besides, we may also compare different baselines more clearly when small-scale codebooks are used. Particularly, the jointly learning methods (Poeem, JPQ, RepCONC, MoPQ) achieve more significant and consistent advantages against the conventional methods (PQ, OPQ); and the methods on retrieval loss minimization (JPQ, RepCONC, MoPQ) are superior than the one minimizing the reconstruction loss (Poeem). Finally, Distill-VQ’s...
relative advantages gradually diminish; when codebooks become sufficiently large, some of the baselines (IQP, RepCONC, MoPQ) are able to get close to Distill-VQ in terms of Recall. This is because larger codebooks are more expressive than the smaller ones, which makes it easier to preserve the dense embeddings’ retrieval performance. However, Distill-VQ still maintains notable advantages in MRR@10 (more discriminative than Recall@100), indicating that the corresponding VQ is still much better than other baselines.

4.2.5 Efficiency and retrieval quality. We further analyze Distill-VQ’s retrieval quality at different working efficiency (Figure 3). The memory efficiency is measured by the bit-rate of each document; e.g., 64-bit is corresponding to the 8x8 codebooks, which leads to a 192x compression of the original dim-768 float-32 document embeddings. The time efficiency is measured by the latency of getting the Top-100 documents for each query. The evaluation is performed on top of FAISS, where the vector quantization results from Distill-VQ are loaded into the IVFOPQ index in FAISS. The original IVFOPQ from FAISS is introduced for comparison.

In Figure 3 I-IV, Recall@100 is measured at different time and memory cost by changing the codebook size (correlation between recall and codebook size is discussed in Table 6). In Figure 3 V-VIII, Recall@100 is measured at different time cost by changing #Search: the amount of posting lists to search (10,000 deployed in total), and #Posting: the number of posting lists to deploy, with the Top 1% to be searched (correlations between recall and both factors are discussed in Table 4, 5). It can be observed that Distill-VQ outperforms IVFOPQ in all of these settings. Besides, the advantages become much more significant at small bit-rates and time latency, reflecting Distill-VQ’s importance in massive-scale scenarios where high compression and acceleration ratios are needed.

4.2.6 Summary of findings. The major findings of the experiments are summarized into the following points.

- Distill-VQ notably outperforms the existing vector quantization methods on MS MARCO and Natural Questions.
- Both IVF and PQ modules may benefit from Distill-VQ: the resulted advantages are more significant when larger compression and acceleration ratios are adopted.
- The knowledge distillation can be effectively performed based on ListNet (enforcing ranking order invariance) and documents sampled from Top-K-In-Batch (w/o using labeled data).
- Distill-VQ results in better efficiency-recall trade-off as compared with the original IVFOPQ in FAISS library.

5 CONCLUSION

We present a novel framework Distill-VQ, which jointly learns IVF and PQ for the optimization of their retrieval performances. In our framework, the PQ modules are learned by distilling knowledge from the well-trained dense embeddings, which enables massive-scale of unlabeled data to be exploited for the improvement of VQ’s performance. Our empirical explorations indicate that Distill-VQ can be effectively conducted by enforcing teachers-students ranking order invariance (ListNet) over documents sampled from unlabeled data (Top-K-In-Batch). Knowing that the proposed method may conveniently work with those off-the-shelf embeddings and easily be integrated with FAISS, we expect it to serve as a generic tool to facilitate high-quality and efficient document retrieval.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Nos. U1936104, 62192784, 62022077 and 61976198) and CCF-Baidu Open Fund.

REFERENCES

[1] Artem Babenko and Victor Lempitsky. 2014. The inverted multi-index. IEEE transactions on pattern analysis and machine intelligence 37, 6 (2014), 1247–1260.
[2] Dmitry Baranchuk, Artem Babenko, and Yury Malkov. 2018. Revisiting the inverted indices for billion-scale approximate nearest neighbors. In ECCV. 202–216.
[3] 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.
[4] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In Proceedings of the 24th international conference on Machine learning. 129–136.
[5] Wei-Cheng Chang, Felix X Xu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. arXiv preprint arXiv:2002.09392 (2020).
[6] Qi Chen, Bing Zhao, Haidong Wang, Minqin Li, Chunanie Liu, Lenzheqiang Li, Yao Yang, and Jingdong Yang. 2021. SPANN: Highly-efficient Billion-scale Approximate Nearest Neighbor Search. arXiv preprint arXiv:2111.08566 (2021).
[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. 2020. Big self-supervised models are strong semi-supervised learners. arXiv preprint arXiv:2008.10029 (2020).
[8] 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.
[9] 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).
[10] Luyu Gao and Jamie Callan. 2021. Unsupervised corpus aware language model pre-training for dense passage retrieval. arXiv preprint arXiv:2108.05140 (2021).
[11] Tiezhen Ge, Kaining He, Qifa Ke, and Jian Sun. 2013. Optimized product quantization. IEEE transactions on pattern analysis and machine intelligence 36, 4 (2013), 744–755.
[12] Daniel Gillick, Sayali Kulkarni, Larry Lassing, Alessandro Presta, Jason Baldrudge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. arXiv preprint arXiv:1909.10506 (2019).
[13] Ruqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chen, and Sanjiv Kumar. 2020. Accelerating large-scale inference with amnesic vector quantization. In International Conference on Machine Learning. PMLR. 3887–3896.
[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015).
[15] Sebastian Hofstätter, Sophía Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving efficient neural ranking models with cross-architecture knowledge distillation. arXiv preprint arXiv:2010.02666 (2020).
[16] Sebastian Hofstätter, Sheng-Chieh Lin, Ji Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. arXiv preprint arXiv:2104.06967 (2021).
[17] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pritoni, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2553–2561.
[18] Hervé 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] Hervé Jegou, Romain Tavenard, Matthijs Douze, and Laurent Amsaleg. 2011. Searching in one billion vectors: re-rank with source coding. In ICASSP. IEEE, 861–864.
[20] Gangwhee Jiang, Hao Wang, Jin Chen, Haoyu Wang, Defu Liu, and Enhong Chen. 2021. xLightFM: Extremely Memory-Efficient Factorization Machine. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 337–346.
[21] Jingyi Liu, Yichun Yin, Liqiang Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. arXiv preprint arXiv:1909.10351 (2019).
[22] Jeff Johnson, Matthijs Douze, and Hervé Jegou. 2019. Billion-scale similarity search with gpus. IEEE Transactions on Big Data (2019).
[23] 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 EMNLP. 6769–6781. https://doi.org/10.3115/v1/D14-1181

[24] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In EMNLP. Association for Computational Linguistics, Doha, Qatar, 1746–1751. https://doi.org/10.3115/v1/D14-1181

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

[26] Chaohuou Li, Soehren Pang, Yunming Liu, Hao Sun, Zheng Liu, Xing Xie, Tianqi Yang, Yanling Cui, Liangjie Zhang, and Qi Zhang. 2021. AdiGNN: Behavior-Graph Augmented Relevance Modeling in Sponsored Search. arXiv preprint arXiv:2104.12080 (2021).

[27] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. arXiv preprint arXiv:1908.09355 (2019).

[28] Suhas Jayaram Subramanya, Rohan Kadekodi, Ravishankar Krishnaswamy, and Harsha Vardhan Simhadri. 2019. Diskank: Fast accurate billion-point nearest neighbor search on a single node. In Proceedings of the 33rd International Conference on Neural Information Processing Systems. 13766–13776.

[29] Defu Lian, Zheng Liu, Weihao Han, Jianjun Zhang, Chaozhuo Li, Yingxia Shao, Defu Liu, Xing Xie, Hao Sun, Denvy Deng, et al. 2022. Progressively Optimized Bi-Granular Document Representation for Scalable Embedding Based Retrieval. In WWW.

[30] Shitao Xiao, Zheng Liu, Weihao Han, Jianjun Zhang, Chaozhuo Li, Yingxia Shao, Defu Liu, Xing Xie, Hao Sun, Denvy Deng, et al. 2022. Training Large-Scale News Recommenders with Pretrained Language Models in the Loop. arXiv e-prints (2021), arXiv:2102.

[31] Shitao Xiao, Zheng Liu, Yingxia Shao, Defu Liu, and Xing Xie. 2021. Matching-oriented Embedding Quantization For Ad-hoc Retrieval. In EMNLP. 8119–8129.

[32] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. 2020. Revisiting knowledge distillation via label smoothing regularization. In ECCV. 3903–3911.

[33] C Yue, M Long, J Wang, Z Han, and Q Wen. 2016. Deep quantization network with noisy student improves imagenet classification. In ECCV. 10687–10698.

[34] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Pang Tang, Jianan Liu, Paul N. Bennett, Janail Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Transformation. In International Conference on Learning Representations. https://openreview.net/forum?id=teFfQgyZ

[35] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. 2020. Revisiting knowledge distillation via label smoothing regularization. In ECCV. 3903–3911.

[36] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. arXiv preprint arXiv:1908.09355 (2019).

[37] Shitao Xiao, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Jointly optimizing query encoder and product quantization to improve retrieval performance. In CIKM. 2487–2496.

[38] 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 CIKM. 2487–2496.

[39] Zheng Liu, Weihao Han, Jianjun Zhang, Chaozhuo Li, Yingxia Shao, Defu Liu, Xing Xie, Hao Sun, Denvy Deng, et al. 2022. Progressively Optimized Bi-Granular Document Representation for Scalable Embedding Based Retrieval. In WWW.