Balanced Knowledge Distillation with Contrastive Learning for Document Re-ranking

Yingrui Yang
Dept. of Computer Science
University of California
Santa Barbara, California, USA
yingruiyang@cs.ucsb.edu

Shanxiu He
Dept. of Computer Science
University of California
Santa Barbara, California, USA
shanxiuhe@cs.ucsb.edu

Yifan Qiao
Dept. of Computer Science
University of California
Santa Barbara, California, USA
yifanqiao@cs.ucsb.edu

Wentai Xie
Dept. of Computer Science
University of California
Santa Barbara, California, USA
wentai@cs.ucsb.edu

Tao Yang
Dept. of Computer Science
University of California
Santa Barbara, California, USA
tyang@cs.ucsb.edu

Abstract

Knowledge distillation is commonly used in training a neural document ranking model by employing a teacher to guide model refinement. As a teacher may not be correct in all cases, over-calibration between the student and teacher models can make training less effective. This paper focuses on the KL divergence loss used for knowledge distillation in document re-ranking, and re-visits balancing of knowledge distillation with explicit contrastive learning. The proposed loss function takes a conservative approach in imitating teacher's behavior, and allows student to deviate from a teacher's model sometimes through training. This paper presents analytic results with an evaluation on MS MARCO passages to validate the usefulness of the proposed loss for the transformer-based ColBERT re-ranking.

CCS Concepts

• **Information systems → Learning to rank**.

Keywords

Neural document ranking, knowledge distillation, KL divergence

ACM Reference Format:
Yingrui Yang, Shanxiu He, Yifan Qiao, Wentai Xie, and Tao Yang. 2023. Balanced Knowledge Distillation with Contrastive Learning for Document Re-ranking. In *Proceedings of the 2023 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '23), July 23, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3578337.3605120

1 Introduction

In practical applications, large-scale search systems for text documents typically utilize a multi-stage ranking process. In the first stage, a fast and relatively simple ranking method is employed to retrieve the top candidate documents that match a query from a large search index. Subsequently, a more complex machine learning algorithm is used to thoroughly re-rank the top results in the second or later stage. For the retrieval stage, recent studies on retrievers that use inverted indices have utilized learned sparse neural representations [7, 9, 11, 14, 28, 32]. Alternatively, dense retrieval employs a dual encoder architecture to produce dense document representations [12, 29, 39, 50, 52, 53]. In terms of re-ranking efforts, although a cross-encoder neural architecture such as BERT [8] can deliver strong relevance performance [39, 46, 54], various efforts have been made to reduce the time complexity of transformer-based ranking [3, 13, 17, 19, 30, 31, 35, 49]. For example, ColBERT [22, 41], a dual encoder architecture with multi-vector document representations, has been well-received for delivering good relevance performance while maintaining reasonable efficiency. The retrieval and re-ranking studies mentioned above have employed contrastive learning to train neural models using positive and negative examples. To improve the relevance of a less complex but more efficient model, knowledge distillation [16] has become increasingly important to transfer knowledge from a powerful teacher model through behavior imitation [13, 17, 29].

This paper is focused on the improvement of the KL divergence loss used for knowledge distillation in improving the ColBERT based model for document re-ranking. A key weakness of using KL divergence is that since a selected teacher model may not perform well in all cases, over-calibration between the student and teacher models with a tight distribution matching may degrade effectiveness. The previous work has used the sum of a log-likelihood based contrastive loss and KL divergence as a weighted regularization to reduce overfitting. Our evaluation with MS MARCO passages finds that this approach does not bring visible benefits in balancing the KL divergence loss in our context for document re-ranking and it may actually degrade the overall relevance in the tested cases. Our analysis shows a key reason is that this approach always aggressively or exactly follows the KL divergence loss in computing the gradient contribution of documents for given a training query, independent of relative performance of the teacher's and student's models.

This paper addresses the aforementioned drawback by proposing an alternative contrastive loss function, integrated with the KL divergence loss to balance knowledge distillation more effectively with explicit contrastive learning. Through analysis, we demonstrate that our proposed contrastive loss can adjust the learning

ICTIR '23, July 23, 2023, Taipei, Taiwan

ACM ISBN 979-8-4007-0073-6/23/07.
https://doi.org/10.1145/3578337.3605120

behavior from a teacher model based on its relative performance compared to a student's model. Overall speaking, it takes a more conservative and adaptive approach compared to the likelihood loss, resulting in improved performance on the average.

Our evaluation with MS MARCO passage ranking shows that a student ColBERT model refined with the proposed loss function could achieve a visibly better relevance score and can outperform several state-of-the-art baselines. This loss function performs reasonably well on TREC deep-learning track test sets.

2 Related Work

Document Re-ranking. After the initial retrieval stage, re-ranking of the top k documents can be achieved through transformer-based neural methods including [6, 8, 27, 31]. For example, RocketQAv2 [39], AR2 [54], and SimLM [46] employ expensive and time-consuming cross-encoder re-rankers to achieve impressive MRR@10 scores of up to 0.437 for MS MARCO passage ranking. To reduce the time complexity of transformer-based re-ranking, various efforts have been made including architecture simplification [19, 35], early exiting [49], and model distillation [3, 13, 17]. Among these, ColBERT [22] has gained popularity due to its dual-encoder architecture and multi-vector document representation, which delivers state-of-the-art relevance on a number of information retrieval benchmarks while maintaining reasonable computational efficiency. Additional studies have been conducted to improve the efficiency of ColBERT re-ranking through clustering and quantization [41, 51]. Our work follows the ColBERT-based multi-vector representation in evaluating the usefulness of the proposed loss in re-ranking. Other studies that advocate multi-vector representations compared to a single-vector representation include ME-BERT [30], ALIGNER [37], and CITADEL [26]. These studies have proposed optimization methods for multi-vector representations and they are orthogonal to our work.

The first-stage retrieval. The evaluation of our re-ranking work uses a SPLADE sparse learned neural model [9, 11] for the first-stage retrieval because it can deliver a strong relevance score with good time and space efficiency during inference based on an inverted index. Its inverted index implementation is fairly efficient on a CPU server without GPU support and can be made even faster with some recently developed optimization [23, 33, 38]. Other learned sparse representations with neural term weights include DeepCT [7], uniCOIL [14, 28] and DeepImpact [32].

Dense retrieval with a single-vector representation and a dual-encoder architecture is an alternative solution for the first-stage retrieval. Although a recent dense retriever (e.g. RocketQAv2 [39], AR2 [54], SimLM [46] and RetroMAE [48]) has achieved a high MRR number, their test setting requires exhaustive exact search to achieve such a number, which needs significantly more computing resource such as GPU. When approximated nearest neighbor (ANN) search [21] is adopted to reduce retrieval cost, the relevance number can drop significantly, as shown in some recent studies [24, 47]. Thus re-ranking may also be applied to boost relevance after dense retrieval with approximated nearest neighbor search.

Knowledge distillation and KL divergence. The concept of knowledge distillation was originally introduced by Hinton et al. [16] to enhance the performance of a simpler neural network with guidance from a teacher model in classification tasks. Subsequently, additional improvements such as layer distillation [44] and student-teacher collaboration [55] have been made. Knowledge distillation has also been shown to be effective in neural document ranking, with the KL-divergence loss becoming a popular choice in recent studies, including TCT-ColBERT [29], ColBERTv2 [41], RocketQAv2 [39], SimLM [46] and RetroMAE [48]. While the Margin-MSE loss proposed by Hofstätter et al. [17] is used in SPLADE-based retrievers [9, 10], our findings suggest that incorporating KL-divergence is still effective for ColBERT re-ranking. Recent studies in dense retrieval [47, 54, 54] have also adopted KL-divergence in their loss functions. Therefore, this paper uses KL-divergence to distill knowledge from a teacher model for re-ranking.

3 Problem Definition and Notations

Symbol	Definition
Q	A query
d_i	A document
$\mathcal{D}^+, \mathcal{D}^-$	Positive/negative document subsets for Q
$\lvert.\rvert$	The size of a set
Θ	Parameters of a scoring model
$S(Q, d_i, \Theta)$	The rank score of a document for a query based on a model with parameters Θ.
p_i	Teacher's top one probability of document d_i, standing for $P(d_i \lvert Q, \mathcal{D}^+, \mathcal{D}^-, \Theta)$
q_i	Student's top one probability of document d_i
λ	Weight hyper-parameter for the contrastive loss
g_A	Stand for $g_A(\lambda, p_i, q_i)$. Relative gradient contribution ratio from document d_i between loss L_A and loss L_{KL}. $\frac{\partial L_A(i)}{\partial q_i} = g_A(\lambda, p_i, q_i) \frac{\partial L_{KL}(i)}{\partial q_i}$
L_{KL}	KL divergence loss
L_{KLL}	KL divergence loss added with the negative log likelihood
L_{BKL}	Proposed loss to balance KL divergence

Table 1: Table of Symbols

Problem definition. Given query Q for searching a collection of N text documents (i.e., $\mathcal{D} = \{d_i\}_{i=1}^{N}$), we obtain the top relevant documents from $\mathcal{D}$ through two search stages. Given top k documents fetched by the first stage retriever, the second stage of search re-ranks top k results from the retrieval stage. While other ranking formulas are possible, this paper mainly follows the ColBERT's Max-Sim formula defined below. Each document d and given query Q use a multi-vector representation $M(d, \Theta)$ and $M(Q, \Theta)$ respectively. Here Θ is the vector of neural parameters involved. $S(Q, d_i, \Theta)$ is a rank scoring function defined as:

$$S(Q, d_i, \Theta) = \sum_{h_i \in M(Q,\Theta)} max_{h_j \in M(d,\Theta)} H(h_i)^T H(h_j). \qquad (1)$$

where h_i, h_j are BERT last layer's embeddings and $H(.)$ is one linear layer with normalization on the output representation.

Training and loss function. For re-ranker training, contrastive learning is widely used, and each query Q in a training dataset comes with one or a few positive documents while negative documents are sampled or selected using various methods. Let $\mathcal{D}_Q^+$ and $\mathcal{D}_Q^-$ be a subset containing all positive and negative documents, respectively. For presentation simplicity, we omit the subscript 'Q' for a specific query in the remaining as $\mathcal{D}^+$ and $\mathcal{D}^-$.

Then, relevance scoring is applied to each pair of the query vector and each document vector for the retrieval or re-ranking purpose. Thus, a probability distribution over the positive and documents documents can be defined as

$$P(d_i|Q, \mathcal{D}^+, \mathcal{D}^-, \Theta) = \frac{\exp(S(Q, d_i, \Theta))}{\sum_{j=1}^{N} \exp(S(Q, d_j, \Theta))}.$$

Notice that in Cao et al. [2] the above formula is called the top one probability. For the simplicity of presentation when no confusion is caused, we will not list Θ and Q explicitly in each symbol below and the loss function is specified for each query Q based on parameters Θ under the training documents $\mathcal{D}^+$ and $\mathcal{D}^-$. Let p_i or q_i denote $P(d_i|Q, \mathcal{D}^+, \mathcal{D}^-, \Theta)$ where p_i and q_i refer to the teacher's and student's prediction, respectively.

To train a model with contrastive learning, the loss function used frequently in the previous ranking studies includes the negative log likelihood or its variation:

$$-\sum_{d_j \in \mathcal{D}^+} \log q_j.$$

KL-divergence defined below has become a dominating choice for knowledge distillation to be included in a loss function as seen in the recent ranking studies [39, 41, 43, 46, 54].

$$L_{KL} = \sum_{d_i \in \mathcal{D}^+ \cup \mathcal{D}^-} p_i \ln \frac{p_i}{q_i}$$

where p_i and q_i refer to the teacher and student top one probability for instance d_i in $\mathcal{D}^+$ or $\mathcal{D}^-$, respectively.

It is a common practice that the above two list-wise losses are added together and we call this weighted sum as L_{KLL}:

$$L_{KLL} = \sum_{d_i \in D^+ \cup D^-} p_i \ln \frac{p_i}{q_i} - \lambda \sum_{d_i \in D^+} \log q_i.$$

4 Balanced KL Divergence with Contrastive Learning

4.1 Case study

We present a study that motivates our effort in re-visiting the balancing of knowledge distillation with a contrastive loss. We use a cross-encoder ranking model called MiniLM-l-6-v2 [34] as a teacher and train a ColBERT re-ranker as a student model through KL divergence based distillation after an initial training warm-up on MS MARCO passages. More details on ColBERT training can be found in Section 6. We compare the MRR@10 number of the student and this teacher for each query in the MS MARCO Dev test set before training. The average MRR number is 0.387 for this Dev test set with 6980 queries. Table 2 shows the number of queries that the MRR of this student model is the same (tie), higher (winning), or lower (losing) than that of the teacher in terms of MRR@10. This table shows that the student model outperforms the teacher in 1172 queries among the 6980 queries in the Dev set before KL divergence based distillation starts. Notice that the above test queries are not used for training and MS MARCO provides a separate set of queries for training.

Figure 1 shows how the MRR number of these 1172 test queries changes compared to the warmup checkpoint after distillation using the MS MARCO training dataset guided by loss functions L_{KL}, L_{KLL} with λ=0.01, 0.02, and 0.05, and by a contrastive loss called L_{BKL} proposed below. We categorize the queries according to the relative MRR@10 change before and after training under a KL divergence loss with and without a contrastive loss. For instance, the percentage marked "worse" means that, after distillation with the corresponding loss, the MRR@10 of the student for such queries becomes smaller than that of the warmup model. From Figure 1, training with L_{KL} only improves the student model in 11.8% of these 1172 queries and 50.8% of the queries have a degradation in performance. Training with L_{KLL} under different λ weights does not bring benefits in balancing knowledge distillation with the negative log-likelihood contrastive loss and it actually makes the performance of students worse in general. The above result inspires us to exploit a different contrastive loss for more effective balancing of knowledge distillation. This figure also shows that the proposed loss can yield a positive improvement. More comparisons on these losses can be found in Table 7 of Section 6.

#Queries student wins	#Tie	#Queries student loses
1172	4288	1520

Table 2: Relative MRR@10 performance of the student ColBERT model and a teacher in MS MARCO passage Dev test set before training

The above study shows that a student's model can outperform a teacher's model in many cases during training and for these cases, there is no need for the student's model to mimic the teacher's behavior exactly since it may degrade performance. Thus our design objective is to take a more conservative and adaptive approach in following the KL divergence loss, based on the relative performance of the teacher's and student's models.

4.2 Loss function proposed

With the above study in mind, we propose an alternative contrastive loss function called BKL for balancing KL divergence. This listwise loss for a query combines the negative entropy component of its positive documents and the L1-norm expression of its negative documents. The following formula uses log (log base 2) and ln (the log base e) because it is common to use function ln for the KL-divergence and function log for the entropy expression.

$$L_{BKL} = \sum_{d_i \in D^+ \cup D^-} p_i \ln \frac{p_i}{q_i} + \lambda \left(\sum_{d_i \in D^+} q_i \log q_i + \frac{1}{\ln 2} \sum_{d_i \in D^-} q_i \right).$$

The design consideration of the above loss is explained as follows.

- While the first component with KL divergence accomplishes the goal of knowledge distillation, the second component weighted by a hyperparameter λ achieves contrastive learning directly. The second component has a goal of ranking positive documents higher than the negative documents for each query. As we discuss below, it achieves the lower bound during training for loss minimization when all negative documents are scored as 0 and all positive documents are scored equally.

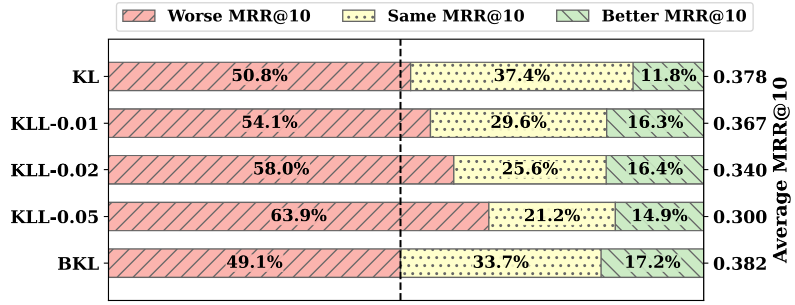

Figure 1: Outcome of training under KL divergence based distillation with and without a contrastive loss for the MS MARCO Dev test query set. The bar distribution shows the percentage in this query subset with better, same or worse MRR@10 numbers by the student model after distillation under the corresponding training loss. The y-axis on the right is the final student MRR@10 average of the 1172 test query subset after training.

The analysis in Section 5 shows that a small λ is preferred. Our evaluation confirms that λ should be chosen small and $\lambda = 0.01$ is a good choice for L_{BKL} in the tested cases.

- It is known that the lower bound of KL-divergence loss is 0 and this is achieved when $\forall d_i, p_i = q_i$. Since the teacher can make mistakes in some query cases, reaching such a lower bound may not make the student model effective. Thus using the above proposed loss, it may not be possible that the lower bounds of both first and second components are reached simultaneously. Thus while the minimization of KL-divergence loss makes the student's scoring as close as possible to the teacher's scoring, the hyperparameter λ provides a balanced use of KL divergence and a contrastive loss.
- Section 5 will provide an additional explanation why L_{BKL} is a better choice than L_{KLL} in balancing KL divergence based knowledge distillation. While L_{KLL} exactly or aggressively follows loss L_{KL} in computing the magnitude and sign of gradient contribution from each document, L_{BKL} is more conservative than L_{KLL} or may deviate from L_{KL} based on the relative performance of teacher and student's models.

4.3 Lower bound analysis

The following analysis shows that this proposed contrastive loss has a constant lower bound for each query which contains a fixed set of positive and negative documents during training. Because this loss function has a lower bound, loss minimization during gradient decent training will have a limit for convergence.

Proposition 4.1. *The lower bound of the BKL loss is* $-\lambda \log s$ *where* $s = |D^+|$.

We describe the proof as follows and explain the design of each component in details.

- It is well known that the lower bound of the KL divergence loss is 0. Thus
$$\sum_{d_i \in D^+ \cup D^-} p_i \ln \frac{p_i}{q_i} \geq 0.$$
This lower bound 0 is achieved when for all documents $d_i \in D^+ \cup D^-$, $p_i = q_i$.
- Next we show that the regularization terms used are bounded by constant $-\lambda \log s$.

Since function $x \log x$ is convex and following Jensen's inequality on a convex function,
$$\frac{\sum_{d_j \in \mathcal{D}^+} q_j \log q_j}{s} \geq \left(\frac{\sum_{d_j \in \mathcal{D}^+} q_j}{s}\right) \log\left(\frac{\sum_{d_j \in \mathcal{D}^+} q_j}{s}\right)$$
where $s = |\mathcal{D}^+|$. The above equality holds when the student's predictions q_j for all positive documents are equal.
Let $z = \sum_{d_i \in \mathcal{D}^-} q_i$. Then
$$\sum_{d_j \in \mathcal{D}^+} q_j \log q_j \geq (1-z) \log\left(\frac{1-z}{s}\right)$$
Then
$$\sum_{d_i \in D^+} p_i \log p_i + \frac{1}{\ln 2} \sum_{d_i \in D^-} q_i \geq (1-z) \log\left(\frac{1-z}{s}\right) + \frac{z}{\ln 2}.$$
The partial derivative of $(1-z)\log(\frac{1-z}{s}) + \frac{z}{\ln 2}$ is $\log \frac{s}{1-z}$ which is positive because $z \in [0, 1]$. Its minimum value is achieved when $z = 0$. Therefore,
$$\lambda\left(\sum_{d_i \in D^+} p_i \log p_i + \frac{1}{\ln 2} \sum_{d_i \in D^-} q_i\right) \geq \lambda \left((1-0) \log \frac{1}{s} + \frac{0}{\ln 2}\right)$$
$$= -\lambda \log s.$$
The lower bound for the second portion of the BKL loss is achieved when all positive documents have the same scoring in the student model. Namely, $q_i = q_j$ for $d_i, d_j \in D^+$. To achieve the lower bound, $z = 0$ as discussed above. Then for all negative documents, $d_i \in D^-$, $d_i = 0$.

5 Relative Gradient Contributions

We explain the usefulness of this new contrastive loss that balances the role of KL-divergence by comparing their gradient contributions from individual documents for parameter update during gradient decent training.

5.1 Gradient contribution from a document

Let θ be one of parameters Θ used in the computation network that maps the input features to score $S(Q, d_i, \Theta)$ for each document d_i as defined in Expressions (1) in Section 2. Then given Loss L_A, and A can be KL, KLL, or BKL,
$$\frac{\partial L_A}{\partial \theta} = \sum_{d_i \in \mathcal{D}^+ \bigcup \mathcal{D}^-} \frac{\partial L_A(i)}{\partial q_i} \frac{\partial q_i}{\partial S(Q, d_i, \Theta)} \frac{\partial S(Q, d_i, \Theta)}{\partial \theta}$$
where $L_A(i)$ is the relevant loss term contributed by document d_i. For loss L_{KL}, $L_{KL}(i) = p_i \ln \frac{p_i}{q_i}$.
$$\frac{\partial L_{KL}(i)}{\partial q_i} = -\frac{p_i}{q_i}.$$

$L_{KLL}(i)$ is $p_i \ln \frac{p_i}{q_i} - \lambda \log q_i$ for a positive document, and $p_i \ln \frac{p_i}{q_i}$ for a negative document.

$$\frac{\partial L_{KLL(i)}}{\partial q_i} = \begin{cases} -\frac{p_i}{q_i} - \frac{\lambda}{q_i} & \text{if } d_i \in \mathcal{D}^+; \\ -\frac{p_i}{q_i} & \text{if } d_i \in \mathcal{D}^-. \end{cases}$$

$L_{BKL}(i)$ is $p_i \ln \frac{p_i}{q_i} + \lambda q_i \log q_i$ for a positive document, and $p_i \ln \frac{p_i}{q_i} + \lambda \frac{q_i}{\ln 2}$ for a negative document.

$$\frac{\partial L_{BKL(i)}}{\partial q_i} =$$

$$\begin{cases} -\frac{p_i}{q_i} + \lambda(\frac{1}{\ln 2} + \log q_i) = -\frac{p_i}{q_i} + \lambda \log(e \times q_i) & \text{if } d_i \in \mathcal{D}^+; \\ -\frac{p_i}{q_i} + \lambda \frac{1}{\ln 2} & \text{if } d_i \in \mathcal{D}^-. \end{cases}$$

To understand the relative gradient ratio between $\frac{\partial L_{KL}}{\partial \theta}$, $\frac{\partial L_{KLL}}{\partial \theta}$, and $\frac{\partial L_{BKL}}{\partial \theta}$, we compare the pairwise ratio of the gradient contribution from document d_i in above additive formulas for $\frac{\partial L_{KL}(i)}{\partial q_i}$. $\frac{\partial L_{KLL}(i)}{\partial q_i}$, and $\frac{\partial L_{BKL}(i)}{\partial q_i}$,

From the above analysis, we can derive the following proposition:

Proposition 5.1.

$$\frac{\partial L_{KLL}(i)}{\partial q_i} = g_{KLL}(\lambda, p_i, q_i) \frac{\partial L_{KL}(i)}{\partial q_i}$$

and

$$\frac{\partial L_{BKL}(i)}{\partial q_i} = g_{BKL}(\lambda, p_i, q_i) \frac{\partial L_{KL}(i)}{\partial q_i}$$

where

$$g_{KLL}(\lambda, p_i, q_i) = \begin{cases} 1 + \frac{\lambda}{p_i} & \text{if } d_i \in \mathcal{D}^+; \\ 1 & \text{if } d_i \in \mathcal{D}^- \end{cases}$$

and

$$g_{BKL}(\lambda, p_i, q_i) = \begin{cases} 1 - \frac{\lambda}{p_i} q_i \log(e \times q_i) & \text{if } d_i \in \mathcal{D}^+; \\ 1 - \frac{q_i \lambda}{p_i \ln 2} & \text{if } d_i \in \mathcal{D}^-. \end{cases}$$

Constant e is the base of the natural logarithms.

5.2 Interpretation of relative contribution ratio

The above analysis reveals the relative gradient contribution ratio between the corresponding terms of KLL and BKL with respect to KL divergence for each document. We discuss the meaning of the sign and magnitude of the gradient contribution ratio g_A for loss L_A below. First as summarized in Table 3, we provide some definition as we classify the update direction of gradient contributed by document d_i in loss L_A compared to loss L_{KL}.

- If $g_A(\lambda, p_i, q_i) > 1$, we call that L_A aggressively follows L_{KL}. The gradient contribution for loss L_A is higher than what would have been with the KL loss with respect to this document.
- If $g_A(\lambda, p_i, q_i) = 1$, we call that loss L_A exactly follows L_{KL}.
- If $0 < g_A(\lambda, p_i, q_i) < 1$, we call that loss L_A conservatively follows L_{KL}.
- If $g_A(\lambda, p_i, q_i) = 0$ or close to 0, the gradient contribution of this document makes 0 or insignificant contribution to the overall gradient. That means a student model does not want to make much weight change based on this document.
- If $g_A(\lambda, p_i, q_i) < 0$, we call that loss L_A deviates from L_{KL}. The gradient contribution for the weight update direction based on loss L_A is opposite to that of the KL divergence for this document.

Condition	Gradient contribution from doc. d_i
$g_A(\lambda, p_i, q_i) > 1$	Loss L_A aggressively follows L_{KL}
$g_A(\lambda, p_i, q_i) = 1$	Loss L_A exactly follows L_{KL}
$0 < g_A(\lambda, p_i, q_i) < 1$	Loss L_A conservatively follows L_{KL}
$g_A(\lambda, p_i, q_i) = 0$ or ≈ 0	No or insig. contribution from d_i in L_A

Table 3: Classification of gradient contribution behavior from document d_i in loss L_A compared to loss L_{KL} during training

It is easy to see that for Loss L_{KLL}, $g_{KLL}(\lambda, p_i, q_i) \geq 1$ for any positive and negative document. Namely L_{KLL} always aggressively or exactly follows L_{KL} independent of relative performance of the teacher and student models. In comparison, the proposed contrastive loss L_{BKL} is much more conservative or cautious than L_{KLL}. The direction of gradient contribution may not be in an agreement with loss L_{KL}.

We use the value ratio or size of teacher prediction p_i and student prediction q_i for document d_i to assess the potential benefits of following the KL divergence loss L_{KL}.

- For positive document d_i, if $p_i \geq q_i$, there is a chance that teacher's model performs better. Student's prediction may be fine as long as the student model ranks document d_i above all negative documents. But if q_i is too small, the chance that q_i is a bad prediction score becomes higher.

  If $p_i < q_i$, there is a chance that student's model performs better while we are not sure about teacher's model.
- Similarly for negative document d_i, if $p_i \geq q_i$, there is a chance that student's model performs better. We are not sure about teacher's prediction.

  If $p_i < q_i$, there is a chance that teacher's model performs better while we are not sure about student's prediction and it may be fine as long as the student's model ranks document d_i below all positive documents.

5.3 Contribution behavior of BKL vs. CKL

Loss L_{BKL} has the following behaviors under different conditions in characterizing the relative size or ratio of teacher's and student's predictions. We summarize them and compare against L_{KLL} in Table 4 where notation g_A stands for $g_A(\lambda, p_i, q_i)$ given Loss A.

- For positive document d_i and when $p_i \geq q_i$, there is a chance that the teacher model might perform better if q_i is too small.
  - if $q_i \leq e^{-1}$, there is a chance that this student might perform poorly. In this case, $g_{BKL}(\lambda, p_i, q_i) \geq 1$ and L_{BKL} exactly or aggressively follows L_{KL}. L_{BKL} is not as aggressive as L_{KLL} and their relative gradient contribution ratio is:
  $$\frac{g_{KLL}(\lambda, p_i, q_i) - 1}{g_{BKL}(\lambda, p_i, q_i) - 1} \geq e^2 \ln 2.$$
  - If $q_i > e^{-1}$, there is a chance that the prediction of this student is acceptable. $1 > g_{BKL}(\lambda, p_i, q_i) \geq 1 - \lambda \log(e)$. L_{BKL} closely follows L_{KL} if λ is chosen to be small.
- For positive document d_i and when $p_i < q_i$, there is a chance that the teacher performs worse.
  - if $q_i \leq e^{-1}$, $g_{BKL}(\lambda, p_i, q_i) \geq 1$ and L_{BKL} exactly or aggressively follows L_{KL}. But L_{BKL} is not as aggressive as L_{KLL} and their relative gradient contribution ratio is: $\frac{g_{KLL}(\lambda, p_i, q_i) - 1}{g_{BKL}(\lambda, p_i, q_i) - 1} \geq e^2 \ln 2$.

Conditions	**Behavior of L_{BKL}**	**Behavior of L_{KLL}**
Positive document d_i		
Student: too low $q_i \leq e^{-1}$	$g_{BKL} \geq 1$. Exactly or aggressively follow L_{KL}. But not as aggressive as L_{KLL}: $\frac{g_{KLL}-1}{g_{BKL}-1} \geq e^2 \ln 2$	$g_{KLL} > 1$. Aggressively follow L_{KL}
Teacher: better $p_i \geq q_i, \quad q_i > e^{-1}$	$1 - \lambda \log(e) \leq g_{BKL} < 1$. Closely follow L_{KL}	
Student: better $p_i < q_i, \quad q_i > e^{-1}$	$g_{BKL} < 1$. Conservatively follow L_{KL} $g_{BKL} < 0$ if $p_i << q_i$. Deviate from L_{KL} if teacher is really bad	
Negative document d_i		
Student: better $p_i \geq q_i$	$0 < g_{BKL} < 1$. Conservatively follow L_{KL}	$g_{KLL} = 1$ Exactly follow L_{KL}
Teacher: better $p_i < q_i$	$0 < g_{BKL} < 1$ if $\frac{p_i}{q_i} > \frac{\lambda}{\ln 2}$. Conservatively follow L_{KL} $g_{BKL} \leq 0$ if $\frac{p_i}{q_i} \leq \frac{\lambda}{\ln 2}$. $\frac{\partial L_{KL}(i)}{\partial q_i}$ & $\frac{\partial L_{BKL}(i)}{\partial q_i}$ are very small with limited impact	

Table 4: A comparison of relative gradient contributions by document d_i in L_{KLL} and L_{BKL} compared to L_{KL}, p_i is teacher prediction, q_i is student prediction. λ is chosen to be small.

  - If $q_i > e^{-1}$, $g_{BKL}(\lambda, p_i, q_i) < 1$. L_{BKL} conservatively follows L_{KL} at most. When p_i becomes very small ($p_i << q_i$) and the teacher performs badly in this case since the prediction for a positive document cannot be too small, $g_{BKL}(\lambda, p_i, q_i) < 0$ and L_{BKL} deviates from L_{KL}. Thus for this subcase, L_{BKL} is expected to outperform L_{KLL}.

- For negative document d_i, and when $p_i \geq q_i$, the student might perform better as we want q_i as small as possible. $0 < g_{BKL}(\lambda, p_i, q_i) < 1$. L_{BKL} conservatively follows L_{KL}. As $g_{KLL}(\lambda, p_i, q_i) = 1$, L_{BKL} is expected to outperform L_{KLL}.
- For negative document d_i, and when $p_i < q_i$, there is a chance that the teacher performs better. If $\frac{p_i}{q_i} > \frac{\lambda}{\ln 2}$, $0 < g_{BKL}(\lambda, p_i, q_i) < 1$, and L_{BKL} still conservatively follows L_{KL}. If $\frac{p_i}{q_i} \leq \frac{\lambda}{\ln 2}$, $g_{BKL}(\lambda, p_i, q_i) \leq 0$.

  When λ is chosen to be small, the chance of having a negative g_{BKL} value becomes small in this case. For example, if $\lambda = 0.01$, $\frac{\lambda}{\ln 2} \approx 0.015$ and then only when $\frac{q_i}{p_i} > 69$, g_{BKL} is negative. In this case, both $\frac{\partial L_{KL}(i)}{\partial q_i}$ and $\frac{\partial L_{BKL}(i)}{\partial q_i}$ are close to 0. The chance of making a negative impact with such a small gradient can be limited.

  In summary, there is still a chance to have a negative g_{BKL} value, which forces us to choose a small λ value. That is the subcase that L_{BKL} does not balance KL divergence very well and there is still room for an improvement in the future.

Overall speaking, the proposed loss L_{BKL} is more conservative than L_{KLL} in following L_{KL} to compute the gradient contribution size and sign. It can differentiate the relative performance of teacher and student's predictions for a given document. That helps to make the update size of parameters relatively smaller in training adaptively. When the performance of teacher is really poor compared to a student for a positive negative document, the direction of gradient contribution in L_{BKL} can be opposite to that of loss L_{KL}, which helps the student to make a better parameter correction during iterative gradient descent update.

6 Evaluation Results

6.1 Datasets and settings

Datasets and metrics. We use the MS MARCO datasets for passage ranking [1, 5]. There are 502,940 training queries, with about 1.1 judgment label per query. The development (Dev) query set is used for test evaluation, and additional test sets include the TREC deep learning (DL) 2019 and 2020 tracks with 97 queries in total and many judgment labels per query. Following the official leader-board standard, for the Dev set of MS MARCO, we report mean reciprocal rank (MRR@10) for relevance instead of using normalized discounted cumulative gain (NDCG) [20] because such a set has about one judgment label per query, which is too sparse to use NDCG. For TREC DL test sets which have many judgment labels per query, we report the commonly used NDCG@10 score. If available, we also list the recall ratio at 1000 which is the percentage of relevant-labeled results appeared in the final top-1000 results.

In all tables below that report our evaluation results in relevance, we perform paired t-tests at the 95% confidence level. In Tables 6, we mark the results with '†' if a baseline result is in statistically significant degradation from our proposed method SPLADE+ColBERT-BKL. In Table 7, '†' is marked for numbers with statistically significant degradation from BKL loss in the last row. We do not perform t-tests on DL'19 and DL'20 test sets as the number of queries in these sets is small.

Training. We follow the settings in ColBERTv2 [41] to train the ColBERT model. Each token embedding has a dimension of 128 as its default size. We adopt MiniLM-l-6-v2 [34] which has been used by ColBERTv2 as its teacher. We use co-Condenser [4] as the pretrained starting checkpoint and adopt sentenceBERT [15] released hard negatives as the negatives used in training.

In terms of training machine resources and parameters, we use a single GPU with 24G memory (A10G) to train ColBERT to converge for up to 20 epochs with a batch size of 32. The resource usage is reasonable compared to what has been used in the previous work [18, 39, 41]. We warm up the model and then switch up the KL, KLL or BKL losses. Learning rates 2e-5 and 1e-5 are used in the warm-up step and the refinement step, respectively.

Dataset	**# Query**	**# Passages**	**Mean Length**	**# Judgments per query**
MS MARCO passage Dev test set	6980	8.8M	67.5	1.1
TREC DL 19 test set	43	–	–	21
TREC DL 20 test set	54	–	–	18

Table 5: Dataset statistics of MS MARCO passages

Model Specs.	Dev		TREC DL19	TREC DL20	
	MRR@10	Recall@1K	NDCG@10	NDCG@10	Distill
Sparse retrieval with inverted indices					
BM25* [40]	$0.172^{\dagger}$	$0.853^{\dagger}$	0.425	0.453	✗
docT5query* [36]	0.277	0.947	0.590	0.597	✗
SPLADE [10]	$0.388^{\dagger}$	0.982	0.714	0.717	✓
Multi-vector retrieval					
ColBERTv2* [41]	0.397	0.984	–	–	✓
CITADEL* [26]	0.399	0.981	0.703	0.702	✓
ALIGNER* [37]	0.403	–	–	–	✗
SLIM^{++*} [25]	0.403	0.968	0.714	0.702	✓
Dual-encoder re-ranker with sparse retrieval					
DeepImpact+ColBERT* [32]	0.362	–	0.722	0.691	✗
uniCOIL+ColBERTv2/CQ* [51]	0.387	0.958	0.746	0.726	✓
SPLADE+ColBERT/BKL	0.407	0.982	0.716	0.736	✓

Table 6: Relevance scores of different one-stage and two-stage algorithms for MS MARCO passage ranking.

		Dev	TREC DL19	TREC DL20
Loss	λ	MRR@10	NDCG@10	NDCG@10
L_{KL}	–	$0.403^{\dagger}$	0.732	0.714
L_{KLL}	0.05	$0.371^{\dagger}$	0.629	0.616
L_{KLL}	0.02	$0.384^{\dagger}$	0.684	0.651
L_{KLL}	0.01	$0.394^{\dagger}$	0.720	0.677
L_{BKL}	0.05	$0.398^{\dagger}$	0.709	0.727
L_{BKL}	0.02	0.406	0.722	0.734
L_{BKL}	0.01	0.407	0.716	0.736

Table 7: Model relevance after refinement with different losses under the same training condition. † is marked for methods where there is a statistically significant performance degradation compared to the BKL loss with $\lambda = 0.01$ at 5%.

6.2 Relevance of different baselines and options

Different methods for re-ranking tasks. Table 6 lists the overall performance of several state-of-the-art baselines from the previous work in multiple categories for MS MARCO passage ranking for searching the Dev test set, and the test sets from TREC'19 and TREC'20 deep learning tracks. We compare them with two-stage search using SPLADE retriever and ColBERT re-ranker trained with BKL. The retriever model follows the training of SPLADE model [10, 23]. If the relevance results of a baseline are copied from its corresponding paper, we tag the reported numbers in the corresponding first entrie with '$*$'. Entry '–' means the result is not available from the corresponding paper. While the results of BKL for DL'19 and DL'20 are reasonable, we mainly compare and discuss MRR@10 numbers of different methods in using the Dev test set of MS MARCO as the DL'19 and DL'20 test sets are too small to achieve appropriate statistical power for t-test.

Table 6 lists a few sparse retrievers as a reference, including BM25 or learned neural representations as their implementation uses invert indices for fast inference without GPU. The SPLADE is retrieval model [10] we use throughout the paper. This table lists the results of several retrieval studies with multi-vector representations including ALIGNER [37], CITADEL [26], and SLIM^{++}[25]. We also list two two-stage search efforts with a learned sparse retriever and ColBERT reranker: DeepImpact [32] and CQ [51]. We can see that ColBERT ranker trained with BKL after SPLADE sparse retrieval achieves 0.407 MRR@10, higher than others listed in this table.

The impact of different loss functions on training. Table 7 lists relevance after model training with BKL or other distillation loss options under the same training condition: fixed negative samples, the same starting warm-up checkpoint, and the same machine. Before training with these losses, the starting point of the ColBERT model has 0.387 MRR@10, which is slightly lower compared to the retrieval performance, indicating that the warmup reranking checkpoint does not add a benefit on top of the SPLADE retrieval model used. "L_{KL}" is the KL-divergence loss without any contrastive loss added. "L_{KLL}" is the negative log likelihood loss added to the KL-divergence loss and λ is chosen as 0.01, 0.02, or 0.05. "L_{BKL}" is the proposed loss added to the KL-divergence loss with the same weight parameter choices: 0.01, 0.02, or 0.05.

From Table 7 we can observe that the ColBERT model trained with loss L_{BKL} with $\lambda = 0.01$ can reach an MRR@10 number higher than the other loss options or settings on the MS MARCO Dev set, and most of these relative gains are statistically significant at the 95% confidence level. A large λ such as $\lambda = 0.05$ or higher does not yield a better performance, which is consistent with our analysis in

Section 5. The result for DL'19 and DL'20 using L_{BKL} with $\lambda = 0.01$ or 0.02 is nearly on par with L_{KL} on average. As mentioned earlier, we view the Dev set performance improvement relatively more important than the DL'19 and DL'20 sets because of the test set size difference.

One can observe that a small increase of λ with KLL loss causes a large performance degradation. When λ is 0.05, the performance drops to 0.371 MRR@10. The sensitivity of KLL performance to the size of λ can be explained based on the relative contribution ratio formula $g_{KLL} = 1 + \frac{\lambda}{p_i}$ for positive documents discussed in Section 5. When p_i value is small for some positive documents scored by the teacher's model, the gradient contribution by L_{KLL} follows the KL divergence loss too aggressively. In this case, there is a good chance that the teacher makes a mistake in scoring. With all three λ choices, L_{KLL} does not bring visible benefits in balancing the KL divergence loss in our context and it actually degrades the overall relevance effectiveness compared to L_{KL}.

7 Concluding Remarks

The contribution of this paper is an alternative contrastive loss for balanced knowledge distillation based on KL-divergence and an evaluation for a ColBERT re-ranker. Our analysis provides an analytical justification to explain that overall speaking, it is more adaptive to the relative performance of teacher's and student's document scoring during model imitation and exhibits a conservative and generally-better learning behavior for the most part compared to the log likelihood based balancing for KL divergence.

Our evaluation is focused on ColBERT re-ranking which can be important for large-scale search with multi-stage search. The BKL refinement on ColBERT re-ranking increases MRR@10 from 0.387 to 0.407 for the MS MARCO passage Dev test set, and outperforms other loss function options. Although the MRR number improvement from some other methods compared is modest, achieving such an increase for this public ranking task is known to be very hard. This illustrates the usefulness of BKL in making a re-ranker more competitive in relevance.

Our future work is to further improve the proposed method in balancing KL divergence as discussed in Sections 5 and 6. The other future work is to study its zero-shot ranking performance and investigate its use in training other search models.

Acknowledgments. We thank anonymous referees for their valuable comments. This work is supported in part by NSF IIS-2225942 and has used the computing resource of the XSEDE and ACCESS programs supported by NSF. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] Daniel Fernando Campos, Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, Li Deng, and Bhaskar Mitra. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *ArXiv* abs/1611.09268 (2016).

[2] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *ICML '07*.

[3] Xuanang Chen, B. He, Kai Hui, L. Sun, and Yingfei Sun. 2020. Simplified TinyBERT: Knowledge Distillation for Document Retrieval. *ArXiv* abs/2009.07531 (2020).

[4] Co-Condenser. 2021. https://huggingface.co/Luyu/co-condenser-marco. (2021).

[5] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Fernando Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2020 Deep Learning Track. *ArXiv* abs/2102.07662 (2020).

[6] Zhuyun Dai and J. Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. *SIGIR* (2019).

[7] Zhuyun Dai and Jamie Callan. 2020. Context-Aware Term Weighting For First Stage Passage Retrieval. *SIGIR* (2020).

[8] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

[9] Thibault Formal, C. Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval. *ArXiv* abs/2109.10086 (2021).

[10] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From Distillation to Hard Negative Sampling: Making Sparse Neural IR Models More Effective. *SIGIR* (2022).

[11] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking. *SIGIR* (2021).

[12] Luyu Gao and Jamie Callan. 2021. Condenser: a Pre-training Architecture for Dense Retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. ACM, Online and Punta Cana, Dominican Republic, 981–993. https://doi.org/10.18653/v1/2021.emnlp-main.75

[13] Luyu Gao, Zhuyun Dai, and J. Callan. 2020. Understanding BERT Rankers Under Distillation. *Proceedings of SIGIR* (2020).

[14] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List. *NAACL* (2021).

[15] MS-MARCO Hard-Negatives. 2022. https://huggingface.co/datasets/sentence-transformers/msmarco-hard-negatives. (2022).

[16] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. *ArXiv* abs/1503.02531 (2015).

[17] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation. *ArXiv* abs/2010.02666 (2020).

[18] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy J. Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. *SIGIR* (2021).

[19] Sebastian Hofstätter, Markus Zlabinger, and A. Hanbury. 2020. Interpretable & Time-Budget-Constrained Contextualization for Re-Ranking. In *ECAI*.

[20] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

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

[22] O. Khattab and Matei A. Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. *SIGIR* (2020).

[23] Carlos Lassance and Stéphane Clinchant. 2022. An Efficiency Study for SPLADE Models. *SIGIR* (2022).

[24] Patrick Lewis, Barlas Oğuz, Wenhan Xiong, Fabio Petroni, Wen tau Yih, and Sebastian Riedel. 2022. Boosted Dense Retriever. *NAACL* (2022).

[25] Minghan Li, Sheng-Chieh Lin, Xueguang Ma, and Jimmy Lin. 2023. SLIM: Sparsified Late Interaction for Multi-Vector Retrieval with Inverted Indexes. In *SIGIR*. arxiv and sigir 2023:2302.06587 [cs.IR]

[26] Minghan Li, Sheng-Chieh Lin, Barlas Oguz, Asish Ghoshal, Jimmy Lin, Yashar Mehdad, Wen tau Yih, and Xilun Chen. 2022. CITADEL: Conditional Token Interaction via Dynamic Lexical Routing for Efficient and Effective Multi-Vector Retrieval. *arXiv 2211.01267* (2022). arXiv:2211.10411 [cs.IR]

[27] Jimmy Lin, Rodrigo Nogueira, and A. Yates. 2020. Pretrained Transformers for Text Ranking: BERT and Beyond. *ArXiv* abs/2010.06467 (2020).

[28] Jimmy J. Lin and Xueguang Ma. 2021. A Few Brief Notes on DeepImpact, COIL, and a Conceptual Framework for Information Retrieval Techniques. *ArXiv* abs/2106.14807 (2021).

[29] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy J. Lin. 2021. In-Batch Negatives for Knowledge Distillation with Tightly-Coupled Teachers for Dense Retrieval. In *REPL4NLP*.

[30] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and M. Collins. 2021. Sparse, Dense, and Attentional Representations for Text Retrieval. *TACL* (2021).

[31] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized Embeddings for Document Ranking. *SIGIR* (2019).

[32] Antonio Mallia, O. Khattab, Nicola Tonellotto, and Torsten Suel. 2021. Learning Passage Impacts for Inverted Indexes. *SIGIR* (2021).

[33] Antonio Mallia, Joel Mackenzie, Torsten Suel, and Nicola Tonellotto. 2022. Faster Learned Sparse Retrieval with Guided Traversal. In *SIGIR*. 1901–1905.

[34] MiniLM-L-6-v2. 2022. https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2. (2022).

[35] Bhaskar Mitra, Sebastian Hofstätter, Hamed Zamani, and Nick Craswell. 2021. Conformer-Kernel with Query Term Independence for Document Retrieval. *SIGIR* (2021).

[36] Rodrigo Nogueira, Wei Yang, Jimmy J. Lin, and Kyunghyun Cho. 2019. Document Expansion by Query Prediction. *ArXiv* abs/1904.08375 (2019).

[37] Yujie Qian, Jinhyuk Lee, Sai Meher Karthik Duddu, Zhuyun Dai, Siddhartha Brahma, Iftekhar Naim, Tao Lei, and Vincent Y. Zhao. 2022. Multi-Vector Retrieval as Sparse Alignment. *arXiv 2211.01267* (2022). arXiv:2211.01267 [cs.CL]

[38] Yifan Qiao, Yingrui Yang, Haixin Lin, and Tao Yang. 2023. Optimizing Guided Traversal for Fast Learned Sparse Retrieval. In *ACM Web Conference (WWW '23)*. ACM, Austin, TX, USA.

[39] Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. RocketQAv2: A Joint Training Method for Dense Passage Retrieval and Passage Re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. ACM, Online and Punta Cana, Dominican Republic, 2825–2835.

[40] Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3 (2009), 333–389.

[41] Keshav Santhanam, O. Khattab, Jon Saad-Falcon, Christopher Potts, and Matei A. Zaharia. 2022. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. *NAACL'22* ArXiv abs/2112.01488 (2022).

[42] Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple Entity-Centric Questions Challenge Dense Retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. ACL, 6138–6148.

[43] Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Kai Zhang, and Daxin Jiang. 2022. UnifieR: A Unified Retriever for Large-Scale Retrieval. *ArXiv* abs/2205.11194 (2022).

[44] Wenxian Shi, Yuxuan Song, Hao Zhou, Bohan Li, and Lei Li. 2021. Follow Your Path: a Progressive Method for Knowledge Distillation. In *ECML/PKDD*.

[45] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *NeurIPS*.

[46] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2023. SimLM: Pre-training with Representation Bottleneck for Dense Passage Retrieval. *ACL* (2023).

[47] Shitao Xiao, Zheng Liu, Weihao Han, Jianjin Zhang, Defu Lian, Yeyun Gong, Qi Chen, Fan Yang, Hao Sun, Yingxia Shao, Denvy Deng, Qi Zhang, and Xing Xie. 2022. Distill-VQ: Learning Retrieval Oriented Vector Quantization By Distilling Knowledge from Dense Embeddings. *SIGIR* (2022).

[48] Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. 2022. RetroMAE: Pre-training Retrieval-oriented Transformers via Masked Auto-Encoder. *EMNLP* (2022).

[49] J. Xin, Rodrigo Nogueira, Y. Yu, and Jimmy Lin. 2020. Early Exiting BERT for Efficient Document Ranking. In *SUSTAINLP*.

[50] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *ICLR*.

[51] Yingrui Yang, Yifan Qiao, and Tao Yang. 2022. Compact Token Representations with Contextual Quantization for Efficient Document Re-ranking. In *ACL*.

[52] 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. *CIKM* (2021).

[53] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing Dense Retrieval Model Training with Hard Negatives. *CoRR* abs/2104.08051 (2021). https://arxiv.org/abs/2104.08051

[54] Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2022. Adversarial Retriever-Ranker for dense text retrieval. *ICLR* (2022).

[55] Wangchunshu Zhou, Canwen Xu, and Julian McAuley. 2021. BERT Learns to Teach: Knowledge Distillation with Meta Learning. In *ACL*.