Weighted KL-Divergence for Document Ranking Model Refinement

Yingrui Yang
Department of Computer Science, University of California
Santa Barbara, California, USA

Yifan Qiao
Department of Computer Science, University of California
Santa Barbara, California, USA

Shanxiu He
Department of Computer Science, University of California
Santa Barbara, California, USA

Tao Yang
Department of Computer Science, University of California
Santa Barbara, California, USA

ABSTRACT

Transformer-based retrieval and reranking models for text document search are often refined through knowledge distillation together with contrastive learning. A tight distribution matching between the teacher and student models can be hard as over-calibration may degrade training effectiveness when a teacher does not perform well. This paper contrastively reweights KL divergence terms to prioritize the alignment between a student and a teacher model for proper separation of positive and negative documents. This paper analyzes and evaluates the proposed loss function on the MS MARCO and BEIR datasets to demonstrate its effectiveness in improving the relevance of tested student models.

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**.

KEYWORDS

Neural document ranking, knowledge distillation, efficient two-stage search.

ACM Reference Format:
Yingrui Yang, Yifan Qiao, Shanxiu He, and Tao Yang. 2024. Weighted KL-Divergence for Document Ranking Model Refinement. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3626772.3657946

1 INTRODUCTION

Learned sparse representations [5, 6, 8, 16, 19] , dual-encoder dense models with single vector representation (e.g. [7, 17, 23, 30, 33]) and multi-vector representations (e.g. [12, 14, 15, 22, 24]) are popular neural information retrieval methods. These methods have been developed as simplifications of expensive cross-encoder neural ranking architectures (e.g., BERT [5]) for faster online inference. To boost the relevance of these simplified neural models, knowledge distillation [9] has been shown to be critical to transfer knowledge from a powerful teacher model during training through behavior imitation. KL divergence is a commonly used training loss for knowledge distillation in document ranking [17, 23, 24, 28, 29].

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07.
https://doi.org/10.1145/3626772.3657946

This paper explores opportunities to further improve the KL-divergence loss function based knowledge distillation in document ranking. One limitation of KL-divergence loss in ranking is its demand for a tight distribution match in each document, lacking prioritization. This can lead to over-calibration issues, particularly when the teacher model performs incorrectly. Prior research has incorporated the weighted sum of a contrastive loss with KL divergence as a regularization to mitigate overfitting, including a recent study [31] named BKL to regularize KL divergence with an entropy and L1-norm based loss. This BKL regularization still behaves incorrectly in several cases and our work will improve the KL divergence loss while avoiding the misbehavior of BKL.

The contribution of this paper is to propose an easy-to-implement modification of KL-divergence loss called contrastively-weighted KL divergence (CKL). Instead of following the aforementioned regularization approach, this new formula guides knowledge distillation in ranking by differentiating the role of positive and negative documents in each query through weighting. Specifically it prioritizes important alignments between a student model and a teacher model and allows dynamic weight adjustment for KL-divergence terms based on relative performance of teacher and student's models in scoring a positive or negative document. This paper gives a design justification on weight prioritization choices and provides an evaluation with MS MARCO and BEIR datasets in improving student ranking models for two-stage search and dense retrieval.

2 BACKGROUND

Problem definition. Given query Q, document search on a collection of N text documents (i.e., $\mathcal{D} = \{d_i\}_{i=1}^{N}$) finds top k results, whose ranking primarily determined by query-document similarities. Let $\mathcal{D}^+$ be the subset of all positive documents for query Q, and $\mathcal{D}^-$ be a subset containing all negative documents for this query. The top one probability distribution over these documents is defined as: $P(d_i|Q, \mathcal{D}^+, \mathcal{D}^-, \Theta) = \frac{\exp(S(Q,d_i,\Theta))}{\sum_{j=1}^{N}\exp(S(Q,d_j,\Theta))}$, where Θ is the vector of neural parameters involved. $S(Q, d_i, \Theta)$ is a scoring function that captures the semantic similarity of a document with query. Knowledge distillation is a training methodology that guides the refinement of a neural student model using a teacher model. 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 ranking model, the standard loss function includes the negative log likelihood or its variation: $-\sum_{d_j \in \mathcal{D}^+} \log q_j$. KL-divergence defined below measures a distance between teacher's and student's distributions, and has become a dominating choice for knowledge distillation for ranking model refinement as seen in the recent studies [23, 24, 26, 28, 34]. $L_{KL} = \sum_{d_i \in \mathcal{D}^+ \cup \mathcal{D}^-} p_i \ln \frac{p_i}{q_i}$.

KL divergence does not differentiate the role of positive and negative documents in a training query and forces a student model to closely match the teacher scoring in all documents. In practice, it is not easy for a student to imitate the behavior of a teach model perfectly in all cases. When a teacher does not perform as good as a student, KL divergence still accumulates the discrepancy between the student and the teacher as a loss without detecting and deprioritizing such a case. Thus the key weakness of knowledge distillation is that it lacks a prioritization and can lead to over calibration. While the BKL loss [31] improves this, its regularization formula over-corrects the behavior of KL divergence and incorrectly lets the student model deviate from or follow teacher's ranking score in a wrong learning direction for several important cases as discussed in Section 3.2 (Figure 2).

3 METHOD PROPOSED

3.1 Contrastively-weighted KL-divergence

To address the above limitation of KL divergence loss, our design introduces a weight for each divergence term $p_i \log \frac{p_i}{q_i}$ to explicitly prioritize the separation of positive and negative documents through weight adjustment. We down-weight positive documents ranked high on the top positions, and negative documents ranked low at the bottom positions by a student model. Specifically for a positive document d_j, the goal is to have student score q_j as large as possible towards 1, and thus we use $(1 - q_j)^\gamma$ as the weight. Here γ is a fixed exponent hyperparameter controlling the scale of weight and we set $\gamma \geq 1$. For a negative document d_i, the goal is to have student score q_i as small as possible towards 0, and we use $(q_i)^{\gamma-\beta_i}$ as the weight. The bias term β_i in the exponent adds another control to fine-tune the weight for a negative document. We require $\gamma - \beta_i \geq 1$.

The proposed *contrastively-weighted* KL-divergence (CKL) is:

$$L_{CKL} = \sum_{d_j \in \mathcal{D}^+} (1 - q_j)^\gamma p_j \ln \frac{p_j}{q_j} + \sum_{d_i \in \mathcal{D}^-} (q_i)^{\gamma-\beta_i} p_i \ln \frac{p_i}{q_i}.$$

For negative document d_i, exponent weight bias

$$\beta_i = \alpha \left(\frac{1}{\pi(i)} - \frac{1}{|\mathcal{D}^+|} \sum_{d_j \in \mathcal{D}^+} \frac{1}{\pi(j)} \right).$$

Here $\pi(i)$, $\pi(j)$ are the rank of negative document d_i and positive document d_j respectively. Bias β_i represents the importance of correcting the ranking position of negative document d_i, compared against the harmonic average position of positive documents. The above use of a rank position is motivated by the previous work which considers the relevance gain by swapping two documents in a ranked order, e.g. LambdaMART [1] and CL-DRD [32].

Figure 1 depicts the weight of KL-divergence terms where x axis lists documents in a descending order of their top one probability. Let $s = |\mathcal{D}^+|$ and $m = |\mathcal{D}^-|$. Without loss of generality, let $\mathcal{D}^+ = \{d_1, \cdots, d_s\}$, and $\mathcal{D}^- = \{d_{s+1}, \cdots, d_{s+m}\}$, and let $q_j \geq q_{j+1}$ for all positive examples $1 \leq j < s - 1$ and $q_i \geq q_{i+1}$ for all negative examples $s + 1 \leq i < s + m$.

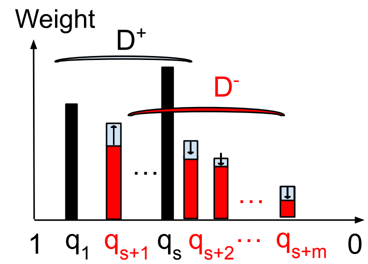

Figure 1: The weights of CKL terms, sorted in a descending order of student's predictions

- For all positive documents d_j where $1 \leq j < s$, because $(1 - q_j)^\gamma \leq (1 - q_{j+1})^\gamma$, a low-scoring positive document is weighted more than a high-scoring positive document, which represents our design priority for the boundary separation of positive and negative documents.
- Among negative documents, assuming $q_i > q_{i+1}$, document d_i is ranked before d_{i+1}. $\frac{1}{\pi(i)} > \frac{1}{\pi(i+1)}$. Thus $\beta_i > \beta_{i+1}$. We have $(q_i)^{\gamma-\beta_i} \geq (q_i)^{\gamma-\beta_{i+1}} \geq (q_{i+1})^{\gamma-\beta_{i+1}}$. High-scoring negative documents are weighted more.
- For any positive and negative document pair d_j and d_i where $1 \leq j \leq s$ and $s + 1 \leq i \leq s + m$, we discuss their relative weight.
  - When $\beta_i \leq 0$, this negative document d_i is ranked lower than the harmonic average position of positive documents. The degree of importance to learn from the teacher in scoring alignment for this document decreases. Thus $(1 - q_j)^\gamma \geq (q_i)^{\gamma-\beta_i}$. Notice $1 - q_j \geq q_i$ and $\gamma \leq \gamma - \beta_i$.
  - When $\beta_i > 0$, negative document d_i is ranked too high compared to the harmonic average position of positive documents. Correcting the rank of this negative document becomes important. CKL upweights the corresponding negative document, and narrows its priority gap to a positive document. The light blue area in each weight bar of a negative document in Figure 1 depicts such an effect with up or down arrows.

3.2 Additional Justifications

We can prove the following properties of CKL.

$$L_{CKL} \geq \sum_{d_j \in \mathcal{D}^+ \cup \mathcal{D}^-} p_j \ln \frac{p_j}{q_j} + \sum_{d_i \in \mathcal{D}^-} p_i (1 - q_i^{\gamma-\beta_i}) \ln q_i + \frac{\gamma}{\log e} \sum_{d_j \in \mathcal{D}^+} q_j \log q_j \tag{1}$$

The first component of the right hand side (RHS) is KL divergence. The third component of RHS is negative entropy of positive documents. The sum of the first and second components in RHS approaches a constant lower bound, reached when $p_i = q_i$ for all positive documents and $q_i = 0$ for all negative documents. The third component of RHS is bounded by $-\frac{2\gamma}{e}$, approached when all q_j values are equal for all positive documents d_j. Thus CKL has a constant lower bound below where p_i values from the teacher's model are a constant.

$$L_{CKL} \geq \sum_{d_i \in \mathcal{D}^-} p_i(-1 + \ln p_i) - \frac{2\gamma}{e}.$$

- **Loss minimization relationship with KL and entropy**. This result shows that minimizing CKL inherently provides a tradeoff between minimizing KL divergence, maximizing entropy of positive documents (does not favor particular positive documents over others) while favoring low scores of all negative documents.
- **Bounded training target.** The above result shows CKL loss has a constant lower bound. If unbounded, training would be unstable and hard to converge. This property indicates CKL preserves the bounded nature as KL-Divergence which has a lower bound 0 when $p_i = q_i$ for all documents d_i.

Relative gradient contributions. Let L be the loss function L where L can be L_{KL}, L_{CKL} or L_{BKL}. $L(i)$ is the loss corresponds to document i. Let θ be a parameter used in the neural network that maps the input features to score $S(Q, d_i, \Theta)$ for each document d_i under loss L. Then

$$\frac{\partial L}{\partial \theta} = \sum_{d_i \in \mathcal{D}^+ \cup \mathcal{D}^-} \frac{\partial L(i)}{\partial q_i} \frac{\partial q_i}{\partial S(Q, d_i, \Theta)} \frac{\partial S(Q, d_i, \Theta)}{\partial \theta},$$

where q_i is the student score of document i. We examine the gradient contribution $\frac{\partial L(i)}{\partial q_i}$ from document d_i in computing $\frac{\partial L}{\partial \theta}$ during the SGD-based training. To understand the relative gradient ratio between $\frac{\partial L_{CKL}}{\partial \theta}$ and $\frac{\partial L_{KL}}{\partial \theta}$, we compare the document-level pairwise ratio of the gradient contribution between CKL and KL losses. $g_{CKL} = \frac{\partial L_{CKL}(i)}{\partial q_i} / \frac{\partial L_{KL}(i)}{\partial q_i}$.

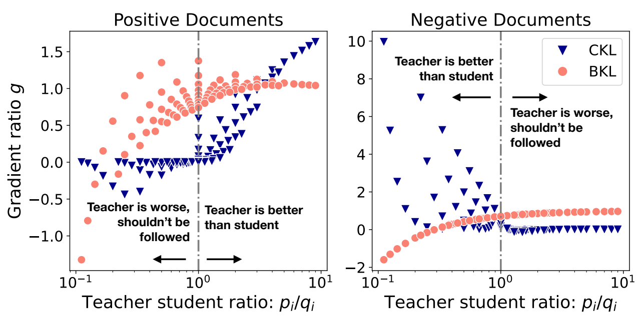

Figure 2: Relative gradient contribution ratio g of CKL in blue triangle and BKL in red bullets

Figure 2 plots the value of gradient contribution ratio g_{CKL} for CKL (g_{CKL}) in a blue triangle, when p_i and q_i vary from 0 to 1. The x-axis is the ratio of teacher student prediction $\frac{p_i}{q_i}$. This figure also depicts relative gradient contribution ratio g_{BKL} of BKL's formula [31] in a red bullet: $g_{BKL} = \frac{\partial L_{BKL}(i)}{\partial q_i} / \frac{\partial L_{KL}(i)}{\partial q_i}$. We fix $\gamma = 5$ and $\beta_i = 0$ in this example for CKL and $\lambda = 0.1$ for BKL for visualization purpose.

Figure 2 illustrates that for positive documents, when the teacher performs better than the student with $\frac{p_i}{q_i} > 1$, $g_{CKL} > 0$ or exceeds 1 and CKL allows the student to follow the teacher's parameter update direction. When the teacher underperforms with $\frac{p_i}{q_i} < 1$, g_{CKL} become close to 0 or even negative, and the student does not learn much from the teacher or its learning deviates from teacher's learning direction. In comparison, BKL still forces the student to follow the teacher's direction with $g_{BKL} > 0$ or even > 1 in most of cases when the teacher is worse.

Similarly for negative documents, when the teacher is better with $\frac{p_i}{q_i} < 1$, $g_{CKL} > 0$ and the student follows the teacher. But g_{BKL} is close to zero and can be negative, meaning the student model does not follow and even incorrectly deviates from teacher. When the teacher is worse with $\frac{p_i}{q_i} > 1$, g_{CKL} is close to 0 and the student does not learn much from the teacher. But in this case, $g_{BKL} > 0$, varying up to 1 and meaning that the student model still follows the teacher, even conservatively.

In summary, CKL's design corrects the misbehavior of BKL by directly weighting KL divergence terms.

Discussion on β_i. Exponent bias β_i for negative document d_i in CKL is updated based on its rank position immediately after each training iteration where q_i is recomputed, which makes the loss function non-differentiable. Thus during training, we opt to periodically update β_i using the latest student's model performance, and the priority adjustment of each negative document is stable for a block of training iterations. This design allows β_i to be treated as a constant in the loss function. This is a reasonable tradeoff as model refinement that addresses ranking accuracy for a negative document takes a number of iterations and continuous β_i adjustment for such a document may not yield sufficient benefits. Hyperparameter α controls the maximum β_i value. We set $0 \leq \alpha \leq \gamma - 1$ which ensures that $\gamma - \beta_i \geq 1$. It is easy to verify that $|\beta_i| \leq \alpha(1 - \frac{1}{|\mathcal{D}^+ \cup \mathcal{D}^-|}) < \alpha$.

4 EVALUATION RESULTS

Evaluation setup. To evaluate in domain performance, we use the MS MARCO dataset with 8.8 million passages [2, 4]. The test query sets include Dev with 6980 queries, and TREC DL 2019/2020 tracks with 43 and 54 queries. To assess the zero-shot performance of a trained student model, we use BEIR which contains 13 publicly available datasets [27], including DBPedia, FiQA, NQ, HotpotQA, NFCorpus, T-COVID, Touche, ArguAna, C-Fever, Fever, Quora, Scidocs, and SciFact. CKL is applied to refine the following two student models: 1) A two-stage search pipeline that combines ColBERT re-ranking with a multi-vector representation [24] and Sparse first-stage SPLADE retriever with a learned neural representation [5]. 2) Dense single-vector retriever SimLM [28]. Two teachers are used: 1) MiniLM-l-6-v2 [21] with 0.407 MRR@10 on MS MARCO Dev on top of SPLADE retrieval 2) a cross encoder from SimLM project [28] with 0.438 MRR@10.

Training steps and configurations. We start training from the officially released pretrained checkpoints. For SPLADE and ColBERT, we follow some of the warm-up settings in SPLADE++ [5] and ColBERTv2 [24]. The cross encoder teacher adopted is MiniLM-l-6-v2 [21], which has been used by ColBERTv2 as its teacher. The warm-up step uses margin-MSE [10] as the loss for knowledge distillation for both SPLADE and ColBERT. The above warm-up step allows the pipeline to deliver 0.399 MRR@10. For model refinement, we use the CKL loss for knowledge distillation or another loss function to compare. We index the corpus using PISA [20]. For SimLM, we compare KL Divergence, CKL or warm up with KL followed by CKL refinement. The teacher we use is the teacher released by the SimLM paper. We use up to four NVIDIA V100 GPUs for model training. Learning rates 2e-5 and 1e-5 are used in the warm-up step and the refinement step, respectively. When training the student retriever, to avoid the expensive re-indexing time during this update, we re-evaluate the top 50 documents per training query as an approximation using the model checkpoint saved after every 2000 batches. The above refinement with CKL for training takes less than 5 epochs to converge.

We use $(\gamma, \alpha) = (5, 1)$ for the ColBERT, SPLADE pipeline and (1,0) for SimLM. When comparing different loss options during training, we maintain the same setup in terms of negative samples, the initial warm-up checkpoint, and the machine environment. For the results presented in this section, we conduct paired t-tests at the 95% confidence level. We denote results that show statistically significant degradation from CKL with '†'. We do not perform t-tests on DL'19 and DL'20 as these sets are relatively small.

	Dev	**DL19**	**DL20**	**BEIR(Avg)**
	MRR@10	NDCG@10	NDCG@10	NDCG@10
SPLADE++ [5]	0.380	0.732	–	0.507
ColBERTv2	0.397	–	–	0.499
SPLADE + top-1000 ColBERT re-ranking				
KLDiv	0.406†	0.716	0.719	0.489
MarginMSE	0.406†	0.704	0.710	0.503
KLDiv_logL	0.405†	0.711	0.699	0.499
CL-DRD	0.406†	0.700	0.693	0.497
BKL	0.407	0.716	0.736	0.506
CKL	**0.411**	**0.744**	**0.741**	**0.515**

Table 1: Two-stage search with different loss options

Two-stage student model SPLADE/ColBERT. Table 1 compares the two-stage search trained under CKL and other distillation loss options in terms of MRR@10 or NDCG@10. The column for BEIR lists the average NDCG@10 across 13 datasets. "KL-Div_logL" is negative log likelihood loss on in-batch negatives plus KL-divergence loss. Other losses include MarginMSE loss [10], CL-DRD [32], and BKL [31]. CKL visibly outperforms other loss options for MS MARCO passages, and also for BEIR zero-shot performance. Overall speaking, CKL delivers a good relevance across the tested datasets. This table also listed published SPLADE++ and ColBERT performance as a reference. Notice that CKL's performance number for MS MARCO and BEIR exceeds or is competitive to several state-of-the-art research studies on multi-vector representations [14, 15, 22].

Model	**Dev**	**DL19**	**DL20**
	MRR@10	NDCG@10	NDCG@10
SimLM w/o title anno.	0.344	0.650	0.641
Trained with KL	0.365	0.685	0.611
Trained with CKL	0.381	0.690	0.696
Trained with KL+CKL	0.391	0.708	0.706

Table 2: Refine dense retriever SimLM with CKL

Student dense retrieval model SimLM. CKL is applied to train on a SOTA dense retrieval model SimLM [28] and Table 2 demonstrates the usefulness of CKL in SimLM. CKL delivers 0.391 in MRR@10 with a warmup using KL divergence. Without warmup, CKL delivers 0.381. For dense retrievers, the released SimLM checkpoint [3] gives 0.344 MRR@10 using the standard MS MARCO. This is below 0.411 reported in [3] which evaluates on the modified MS MARCO dataset with title annotation. Title annotation is considered unfair in [13] since the original dataset released doesn't utilize title information. The numbers reported from recent papers RocketQAv2 [23], LexMAE [25], RetroMAE and RetroMAE-2 [18, 29] were boosted by this title annotation. All experiments for CKL follow the standard approach to use the original MS MARCO without title annotation, and the CKL improvement in refining SimLM is reasonable compared to KL.

Behavior characteristics of CKL. Figure 3 shows several behavior characteristics and differences when using CKL and KL during model refinement of two-stage search with ColBERT/SPLADE, which reflects our design consideration explained in Section 3. Figures 3(a) and (b) depict the difference between the lowest top one probability of positive documents with the highest top one probability of negative documents in top 10 results predicted by a student model during training and DL19/DL20 testing. This result reflects the design objective of CKL loss which tries to separate positive documents from negative documents in terms of probability distribution in each query. Figures 3(c) and (d) show Shannon's entropy among positive documents during training and during DL19 and DL20 testing. This entropy value with CKL is higher than that of KL, reflecting the result of Expression (1): minimizing CKL implies that the entropy among positive documents is maximized.

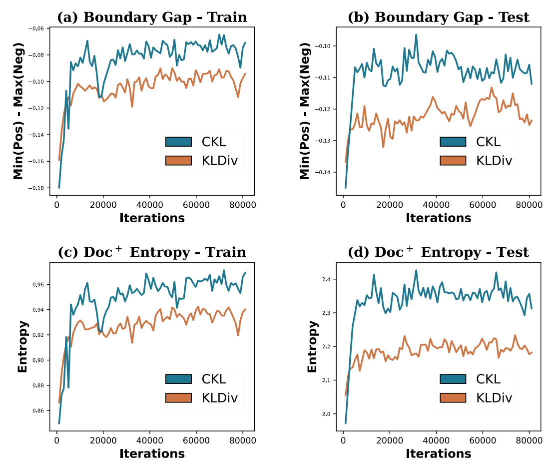

Figure 3: Behavior characteristics of CKL/KL during training

5 CONCLUDING REMARKS

The novelty of this work is an easy-to-implement and effective loss modification of KL-divergence for ranking model refinement with a justification. The applicability of CKL is restricted to training where two-level positive and negative labels are available per query. That is common in practice because it is costly to build a large training dataset for ranking with multi-level labels. The original MS MARCO training dataset does not contain negative documents labeled, and negatives are added algorithmically [11, 23, 30]. The evaluation shows CKL can effectively boost the relevance of tested student models, and achieve reasonably strong relevance numbers compared to other recent ranking studies.

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 ACCESS program 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] Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11, 23-581 (2010), 81.

[2] 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).

[3] SimLM Checkpoint. [n. d.]. https://huggingface.co/intfloat/simlm-base-msmarco-finetuned. ([n. d.]).

[4] 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).

[5] 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).

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

[7] 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

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

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

[10] 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).

[11] V. Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Yu Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. *EMNLP'2020* ArXiv abs/2010.08191 (2020).

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

[13] Carlos Lassance and Stephane Clinchant. 2023. The Tale of Two MSMARCO - and Their Unfair Comparisons. In *Proceed. of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. ACM, New York, NY, USA, 2431–2435.

[14] 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]

[15] 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]

[16] 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).

[17] 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*.

[18] Zheng Liu, Shitao Xiao, Yingxia Shao, and Zhao Cao. 2023. RetroMAE-2: Duplex Masked Auto-Encoder For Pre-Training Retrieval-Oriented Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 2635–2648.

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

[20] Antonio Mallia, Michal Siedlaczek, Joel Mackenzie, and Torsten Suel. 2019. PISA: Performant indexes and search for academia. *Proceedings of the Open-Source IR Replicability Challenge* (2019).

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

[22] 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]

[23] 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.

[24] 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).

[25] Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Xiaolong Huang, Binxing Jiao, Linjun Yang, and Daxin Jiang. 2023. LexMAE: Lexicon-Bottlenecked Pretraining for Large-Scale Retrieval. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=PfpEtB3-csK

[26] 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).

[27] 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*.

[28] 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).

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

[30] 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*.

[31] Yingrui Yang, Shanxiu He, Yifan Qiao, Wentai Xie, and Tao Yang. 2023. Balanced Knowledge Distillation with Contrastive Learning for Document Re-Ranking. In *Proc. of 2023 ACM SIGIR Inter. Conf. on Theory of Information Retrieval* (Taipei, Taiwan) *(ICTIR '23)*. 247–255.

[32] Hansi Zeng, Hamed Zamani, and Vishwa Vinay. 2022. Curriculum Learning for Dense Retrieval Distillation. *SIGIR* (2022).

[33] 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).

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