Squeezing Water from a Stone: A Bag of Tricks for Further Improving Cross-Encoder Effectiveness for Reranking

Pradeep, R.; Liu, Y.; Zhang, X.; Li, Y.; Yates, A.; Lin, J.

DOI
10.1007/978-3-030-99736-6_44

Publication date
2022

Document Version
Final published version

Published in
Advances in Information Retrieval

Squeezing Water from a Stone: A Bag of Tricks for Further Improving Cross-Encoder Effectiveness for Reranking

Ronak Pradeep[1(✉)], Yuqi Liu[1], Xinyu Zhang[1], Yilin Li[1], Andrew Yates[2,3], and Jimmy Lin[1]

[1] University of Waterloo, Waterloo, Canada
rpradeep@uwaterloo.ca
[2] University of Amsterdam, Amsterdam, The Netherlands
[3] Max Planck Institute for Informatics, Saarbrücken, Germany

Abstract. While much recent work has demonstrated that hard negative mining can be used to train better bi-encoder models, few have considered it in the context of cross-encoders, which are key ingredients in modern retrieval pipelines due to their high effectiveness. One noteworthy exception comes from Gao et al. [13], who propose to train cross-encoders by adapting the well-known NCE loss and augmenting it with a "localized" selection of hard negative examples from the first-stage retriever, which they call the Localized Contrastive Estimation (LCE) loss. In this work, we present a replication study of LCE on a different task and combine it with several other "tricks" (e.g., replacing $\text{BERT}_{\text{Base}}$ with $\text{ELECTRA}_{\text{Base}}$ and replacing BM25 with TCT-ColBERTv2) to substantially improve ranking effectiveness. We attempt to more systematically explore certain parts of the hyperparameter space, including the choice of losses and the group size in the LCE loss. While our findings, for the most part, align with those from the original paper, we observe that for MS MARCO passage, orienting the retriever used for hard negative mining with the first-stage retriever used for inference is not as critical for improving effectiveness across all settings. Our code and documentation can be found in: https://github.com/castorini/replicate-lce.

1 Introduction

After the introduction of BERT [6] in October 2018, a simple retrieve-then-rerank approach quickly emerged in January 2019 as an effective method for applying pretrained transformers to passage retrieval [34]. This model, called monoBERT, represents the first instance of what has later become known as cross-encoders for retrieval, a class of reranking models that includes MaxP [4], CEDR [33], Birch [1], PARADE [25], and many others.

R. Pradeep and Y. Liu—Equal contribution.

M. Hagen et al. (Eds.): ECIR 2022, LNCS 13185, pp. 655–670, 2022.
https://doi.org/10.1007/978-3-030-99736-6_44

Innovations in cross-encoder models have of late stagnated in comparison to rapid developments in retrieval models based on learned dense representations [21,46] as well as learned sparse representations [5,8,12]. Part of this excitement stems from the ability of these models to directly perform ranking, as opposed to reranking based on some first-stage retrieval method to generate a list of candidates. However, reranking remains important because the output of even the best dense, sparse, or hybrid retrieval models can be further improved via reranking—and state-of-the-art effectiveness on popular benchmark datasets is achieved only by combining effective first-stage retrieval and reranking in a multi-stage ranking architecture.

Thus, although the attention of most researchers today lies beyond cross-encoders, there remain opportunities for further innovation with this class of models. In this paper, we start with the basic monoBERT model, dating back to January 2019 (which might as well be from the stone age in "neural network time"), and through a series of replication and generalization experiments, are able to improve its effectiveness by nearly 7 points absolute (20% relative) on the popular MS MARCO passage ranking task. We are, in fact, quite surprised that there is still this much effectiveness that could be squeezed out of such a mature model. How did we accomplish this? We describe below:

1. Building on the observations of Zhang et al. [53], we switched the backbone of the cross-encoder to $\text{ELECTRA}_{\text{Base}}$.
2. We replicated and then generalized the findings of Gao et al. [13], confirming the effectiveness of the LCE loss compared to hinge and cross entropy (CE) loss on MS MARCO passage ranking [2], a task not evaluated in the original paper.
3. Leveraging advances in first-stage dense retrieval methods, we used TCT-ColBERTv2 [29] to generate both the first-stage base retrieval runs for reranking and hard negatives for training our cross-encoders.
4. While Gao et al. [13] evaluated various LCE settings with up to 7 negative passages for each positive example in the batch, we extended this to 31 negatives and continued to see improvements in effectiveness.
5. Further generalizing, we noted a surprising result in our replication on MS MARCO passage ranking: it does not seem as critical as described in the original paper to train with negatives that are drawn from the first-stage retriever used for inference. That is, training with BM25 negatives or TCT-ColBERTv2 negatives both result in rerankers that perform comparably when a fixed first-stage retriever is used for reranking, for certain LCE settings. However, for inference, switching a BM25 first-stage retriever out for a TCT-ColBERTv2 first-stage retriever still brings about a significant effectiveness boost.

With the bag of tricks described above, we show that $\text{monoELECTRA}_{\text{Base}}$ can achieve an MRR@10 of 0.414 on the development set of the MS MARCO passage ranking task and an MRR@10 of 0.404 on the (blind) evaluation set. Note that this is accomplished with a standard "base" model size and without the use of any

ensembles. While admittedly, none of these "tricks" in isolation are particularly noteworthy, taken together, they show that there is still room for significant improvements in a basic cross-encoder design that dates from January 2019.

2 Related Work

2.1 Cross-Encoders

As discussed, the first cross-encoder for reranking, monoBERT [34], quickly emerged after the introduction of BERT [6] itself. It followed the approach recommended by the BERT authors to handle (*query, passage*) input pairs, and demonstrated a huge leap in terms of effectiveness on the MS MARCO passage ranking [2] and TREC CAR [7] datasets. While vanilla monoBERT showed great improvement on the passage retrieval task, it was not designed to handle long input sequences as required for document retrieval. A lot of the follow-up BERT-based cross-encoder work [1,4,25,33] attempted to address this issue by either performing multiple inferences on different segments of the document or making additional architectural changes on top of BERT to better handle the longer document text.

In addition to cross-encoders relying on BERT-based pretrained Language Models (pLMs), another genre of cross-encoders takes advantage of the sequence-to-sequence pLM paradigm. Examples of these are monoT5 [35] and duoT5 [40], which use T5 [42], an extensively pretrained encoder-decoder language model. As we mostly focus on BERT-based cross-encoders in this work, we will skip the details and refer interested readers to the original papers.

There exists a strong need for better cross-encoders, which demonstrate state-of-the-art effectiveness in information retrieval tasks in various domains, even in a zero-shot setting [38,39,43,52]. They also form a vital backbone in a wide range of natural language processing tasks, including fact verification [20,37] and question answering [48].

2.2 Bi-Encoders

The success of DPR [21] and ANCE [46] revitalized bi-encoders in the new era of BERT. The goal of a bi-encoder is to learn a transformer-based mapping from queries and documents into dense fixed-width vectors wherein the inner product between the query vector and the relevant document vector is maximized. A lot of work has gone into understanding and better learning such a mapping [9,10,17,29]. A more thorough survey can be found in Lin et al. [27].

Lin et al. [28] train a bi-encoder by using on-the-fly knowledge distillation from a ColBERT [22] teacher model that computes soft-labels for in-batch negatives. This is captured by using the KL-divergence between the score distributions of the student and teacher models for all examples in the batch. They show that using this loss in addition to the standard cross entropy loss over relevance labels results in better scores.

In their follow-up work [29], a "HN+" hard-negative mining strategy is incorporated to further improve their bi-encoder, dubbed "TCT-ColBERTv2". Here, a trained TCT-ColBERT is used to first mine hard negatives to replace the BM25-based negatives. Then, the ColBERT teacher is fine-tuned using these hard negatives and the improved teacher is distilled into the bi-encoder to give TCT-ColBERTv2.

2.3 Hard Negatives

Prior work shows that the selection of negative examples is critical in training bi-encoders. Karpukhin et al. [21] compare the training effectiveness of random negatives, BM25 negatives, and in-batch negatives, and find that a mixture of BM25 and in-batch negatives yields optimal results. Xiong et al. [46] prove theoretically that local negatives are sub-optimal for dense retrieval learning. Then, they propose to prepare global negatives using the current dense retrieval model in parallel with training, which requires periodically re-indexing the corpus and retrieval. Qu et al. [41] also propose to prepare the hard negatives using the current dense retrieval model, but after the training is finished instead of on the fly. However, the paper reports that the hard negatives prepared in this way alone could degrade training and are only effective after being filtered according to an independently trained cross-encoder model. Zhan et al. [51] find that such instability caused by hard negatives could be alleviated by adding random negatives. Additionally, they periodically re-prepare the hard negatives in the ANCE [46] manner, but only update the query encoder to save the re-indexing time. All the above works confirm the importance of hard negatives and show various degrees of effectiveness.

In addition to the work described above, which focuses on hard negative training strategies for DPR-like bi-encoder fine-tuning, other works show similar observations in different methods that aid bi-encoders. Gao et al. [9] find that hard negatives are still crucial when the model is further pretrained in a way to enrich the representation of the `[CLS]` token, which they named Condenser. Its successor, the coCondenser [10] behaves the same after the model is additionally pre–fine-tuned on another corpus-aware unsupervised task. Hard negative mining has also been shown to be important when knowledge distillation is applied on the bi-encoders [17,29].

In contrast to the plenteous studies of hard negatives aiding bi-encoders, we only find Gao et al. [13] successfully incorporating hard negatives in cross-encoder training. To demonstrate the effectiveness of the proposed Localized Contrastive Estimation (LCE) loss, they show that training cross-encoders incorporated with the loss and harder negatives[1] can significantly improve reranking effectiveness, especially when training instances follow the same distribution as the results returned by the first-stage retrievers. Details will be introduced in Sect. 5.2.

[1] Here the "easy" negatives refer to the negatives sampled from BM25 results and the "hard" negatives refer to the ones sampled from HDCT [5] results.

2.4 Pretrained Transformers for Cross-Encoders

Various pretrained models have been proposed after BERT [6]. Most of the works aim at improving the general language representation ability or lowering the pretraining cost [3,24,30]. A few pretrained language models under this line have been compared in Zhang et al. [53] in *ad hoc* retrieval tasks.

Another line of work focuses on improving pLMs with IR-specific pretraining objectives. PROP [32] and B-PROP [31] propose to add a representative words prediction (ROP) task along with MLM in the pretraining stage. To prepare the training data for ROP, a document language model is used to sample a list of "pseudo queries" and their likelihoods. Then the queries are paired as (q^+, q^-) such that q^+ has a higher likelihood than q^-, and the BERT model is further pretrained to score the q^+ higher than the q^-. PROP uses a unigram language model as the document language model while its successor B-PROP uses BERT. Both are tested on a downstream retrieval task by fine-tuning a cross-encoder initialized with the ROP-pretrained BERT instead of BERT with standard pretraining. While Gao et al. [9] also propose an IR-specific pretraining task, it focuses on enriching LMs for the bi-encoder setting.

3 Loss Functions

In this section, we review common loss functions (cross entropy and hinge loss) used in cross-encoder fine-tuning and then describe the Localized Contrastive Estimation (LCE) loss function proposed by Gao et al. [13].

3.1 Cross Entropy and Hinge Loss

We begin with a quick review of how cross-encoders typically compute the relevance score given a query q and a document d, borrowing the formulation from Lin et al. [27]:

$$z_{q,d} = T_{[CLS]}W + b \tag{1}$$

where $z_{q,d}$ is the relevance score of the (query, document) pair, $T_{[CLS]}$ stands for the representation of the `[CLS]` token in the final layer, and W and b are the weight and the bias in the final classification layer. The dimensions of W and b might change according to the loss function—when the model is fine-tuned with the cross entropy loss, $W \in \mathbb{R}^{D\times 2}$ and $b \in \mathbb{R}^2$, whereas when it is fine-tuned with the hinge or the LCE loss, $W \in \mathbb{R}^D$ and $b \in \mathbb{R}$. That is, the output has two dimensions with cross entropy loss, one each for the relevant and non-relevant classes, while with the other two losses, the output has only one dimension, for the relevant class only.

Early cross-encoders fine-tune BERT under a classification task, using the cross entropy loss, as recommended in BERT:

$$s_{q,d} = \mathrm{softmax}(z_{q,d})_1 \tag{2}$$

$$L_{CE} = -\sum \log(s_{q,d^+}) - \sum \log(1 - s_{q,d^-}) \tag{3}$$

where $\text{softmax}(\cdot)_1$ corresponds to the softmax score of the relevant label which by convention is indexed by 1, d^+ indicates a relevant document and d^- indicates a non-relevant document. We will use this notation from now on.

Later, MacAvaney et al. [33] fine-tune cross-encoders with the hinge loss (sometimes called max margin loss), which is more commonly used in pre-BERT neural reranker training [14,19,45]:

$$L_{\text{hinge}} = \max(0, 1 - z_{q,d^+} + z_{q,d^-}) \tag{4}$$

In the literature, cross entropy loss and hinge loss represent the two "basic" ways of training cross-encoders.

3.2 Localized Contrastive Estimation

Gao et al. [13] note that the above cross entropy loss computation considers only one document per batch per query. While not discussed in the original paper, hinge loss is similarly limited by not being able to use multiple negatives per positive unless done in a pairwise independent fashion. They also note it is important that negative examples be true negatives, especially on datasets like MS MARCO passage where many relevant passages remain unlabelled for each query. Gao et al. [13] propose the Localized Contrastive Estimation (LCE) loss to address these issues:

$$L_{LCE_q} := -\log \frac{exp(z_{q,d^+})}{\sum_{d \in G_q} exp(z_{q,d})} \tag{5}$$

$$L_{LCE} := \frac{1}{|Q|} \sum_{q \in Q, G_q \sim R_q^m} L_{LCE_q} \tag{6}$$

where R_q^m is the collection of documents top-ranked by a first-stage retriever for query q, and G_q refers to a group of documents for query q, which consists of a relevant document d^+ and $n-1$ non-relevant documents d^- sampled from R_q^m, where n is the group size.

The LCE loss combines the Noise Contrastive Estimation (NCE) [15] loss (used in, for example, Karpukhin et al. [21]) with "localized" selection of negative examples. The NCE loss scores the positive instance and multiple negative instances, normalizes all of them into probabilities, passing them through the softmax function, and encourages the model to score the positive higher than the negatives. LCE "localizes" this loss by sampling negative training examples from the top-ranked documents produced by the first-stage retriever. In combination, this loss should produce a reranker that succeeds at handling the top-ranked documents specific to a first-stage retriever while also not collapsing to match based on the confounding characteristics in the retriever's hard negative samples.

4 Experimental Setup

In this section, we describe the data and experimental configurations used in our replication. Note that *replication* indicates using a different experimental setup (e.g., implementation, framework, dataset, etc.) to generalize findings from the original paper, whereas *reproduction* indicates verifying the original paper's findings using the same experimental setup.[2]

4.1 Data

We use the MS MARCO *passage* ranking dataset [2] (MS MARCO for later reference), a large-scale *ad hoc* retrieval dataset constructed from the Bing search log. It contains 8.8 million passages and around 800K queries for training, where most of the queries have a single passage labelled relevant. These do not necessarily represent *all* true relevant passages, as it is likely that many queries in the dataset have more than one relevant passage. This setting is often called "sparse labelling". On the evaluation side, there is a small development set with 6980 queries and a blind test set with 6837 queries, both of which are similarly sparsely labelled.

We report MRR@10, the official metric, and Recall@1K (R@1K) on the small development set for all our experiments. Evaluating on the blind test set requires the submission of runs to the organizers' official leaderboard. To avoid probing the test set across various settings, we chose to submit only the test set run produced by the most effective system based on development set scores.

Note that the original work [13] uses the MS MARCO *document* ranking dataset. Thus, our experiments generalize their findings to cover the MS MARCO *passage* ranking dataset and additionally thoroughly explore certain parts in the hyperparameter space.

First-stage rankings (runs) are generated for MS MARCO's training, development, and test query sets with two retrievers: BM25 and TCT-ColBERTv2 [29]. We use the Anserini IR toolkit [47], which is built on Lucene, to generate the BM25 runs. The parameters k_1 and b are found using grid search over the range $[0.6, 1.2]$ and $[0.5, 0.9]$, respectively, both with step size 0.1. The tuning is based on 5 different randomly prepared query subsets, optimizing Recall@1K, following the reproduction documentation in Anserini.[3] We use the Pyserini IR toolkit [26] to generate the TCT-ColBERTv2 runs following the reproduction documentation in Pyserini.[4] We leverage the model trained with the HN+ setting as it optimizes the effectiveness of the primary metrics.

[2] The terms replication and reproduction are used in the sense articulated by the ACM Artifact Review and Badging (v1.1) policy; note that the definitions of the two terms are swapped in Artifact Review and Badging (v1.0).

[3] https://github.com/castorini/anserini/blob/master/docs/experiments-msmarco-passage.md.

[4] https://github.com/castorini/pyserini/blob/master/docs/experiments-tct_colbert-v2.md.

We first retrieve the top-200 passages for each query in the training query set, from which we randomly sample negative examples without replacement following Gao et al. [13]. The method of creating the training set differs from the general approach of cross-encoders which instead just relies on the official small triples training file provided by the organizers.[5] However, such an approach is common in both bi-encoders and cross-encoders when they rely on hard negative sampling.

We retrieve 1K passages for each query in the development set and test set. These form the base first-stage retriever runs which are later reranked by the cross-encoders.

4.2 Training and Inference

Our cross-encoder training and inference experiments are run on Capreolus [49,50], a toolkit for end-to-end neural ad hoc retrieval. We take advantage of its support for the MS MARCO passage ranking task, the monoBERT cross-encoder, and the training, reranking, and inference pipeline. We use the provided hinge and cross entropy loss functions, and incorporate the LCE loss into the toolkit.

The maximum numbers of tokens for the query and the entire input sequence ("`[CLS]` *query* `[SEP]` *passage* `[SEP]`") are set to 50 and 256, respectively. For all experiments, we initialize monoBERT with $\text{ELECTRA}_{\text{Base}}$, using the checkpoint released on HuggingFace [44].[6] We choose $\text{ELECTRA}_{\text{Base}}$ as the starting point for fine-tuning as it appears to be the most stable and effective pLM overall among those considered by Zhang et al. [53].

In all our experiments, we train monoBERT for 300K steps with a batch size of 16. We use the Adam optimizer [23] with a learning rate of $1e{-}5$, apply linear warm-up for the first 30K steps, and apply linear decay following warm-up. All experiments are run on Quadro RTX 8000 GPUs with TensorFlow 2.3.0. We use mixed-precision training in all the experiments.

5 Results and Discussion

In this section, we first compare the results of the three loss functions (hinge, cross entropy, and LCE) when using BM25 and TCT-ColBERTv2 first-stage retrievers. Here, we aim to show that the cross-encoders trained with the LCE loss outperform those with the other two losses on the MS MARCO passage ranking task. Then we compare their effectiveness as we vary both the source of negatives during training and the first-stage retriever during inference. Finally, we show the effect of the group size to confirm the finding that the effectiveness of cross-encoders trained with the LCE loss increases with group size, which also means it increases with more negative samples.

[5] https://msmarco.blob.core.windows.net/msmarcoranking/triples.train.small.tar.gz.
[6] `google/electra-base-discriminator`.

5.1 Loss Functions

Table 1 reports results with different loss functions and first-stage retrievers used during training and inference. In the first block, we report the scores of BM25 and TCT-ColBERTv2, which form the two baseline first-stage retrieval runs we consider for all the Capreolus rerankers.

Table 1. MRR@10 and Recall@1K with different loss functions when using BM25/TCT-ColBERTv2 as the source of hard negative and first-stage runfile. The n in the table indicates the group size. For hinge and LCE, each group always contains a positive example and $n-1$ negative examples. For CE, each group only contains one data point, which could be either a positive or negative example. "–" indicates not applicable or the score was not reported in the original papers. Superscripts indicate significantly higher results ($p < 0.01$ with paired t-tests) after Bonferroni correction, e.g., ^{a} indicates the entry is significantly higher than the results in row (a).

	HN+ First Stage	Loss	n	MRR@10	R@1K
Baselines					
(a) BM25		–	–	0.187	0.857
(b) TCT-ColBERTv2		–	–	0.359	0.969
Prior cross-encoder work					
(c) monoBERT$_{\text{Base}}$ [34]	BM25	CE	1	0.347	–
(d) monoBERT$_{\text{Base}}$ [36]	BM25	CE	1	0.348	–
(e) monoBERT$_{\text{Base}}$ [11]	BM25	CE	1	0.353	–
(f) monoBERT$_{\text{Base}}$ [18]	BM25	CE	1	0.376	–
(g) monoBERT$_{\text{Base}}$ [34]	BM25	CE	1	0.365	–
(h) monoBERT$_{\text{Large}}$ [18]	BM25	CE	1	0.366	–
(i) monoBERT$_{\text{Large}}$ [35]	BM25	CE	1	0.372	–
(j) monoT5$_{\text{Base}}$ [35]	BM25	CE	1	0.381	–
Capreolus cross-encoders					
(1) monoELECTRA$_{\text{Base}}$	BM25	CE	1	0.378^{ab}	0.857
(2) monoELECTRA$_{\text{Base}}$		Hinge	2	0.379^{ab5}	
(3) monoELECTRA$_{\text{Base}}$		LCE	2	0.378^{ab5}	
(4) monoELECTRA$_{\text{Base}}$		LCE	8	$0.391^{ab12356}$	
(5) monoELECTRA$_{\text{Base}}$	TCT-ColBERTv2	CE	1	0.365^{a}	0.969
(6) monoELECTRA$_{\text{Base}}$		Hinge	2	0.375^{ab}	
(7) monoELECTRA$_{\text{Base}}$		LCE	2	$0.393^{ab12356}$	
(8) monoELECTRA$_{\text{Base}}$		LCE	8	$0.401^{ab123456}$	

The second block reports the scores of various comparable cross-encoders from various groups reported in the literature. We copy over the monoBERT and monoT5 scores from their original papers, rows (c), (g), and (j), respectively. We additionally include other monoBERT results reported by different groups because we observe a large variance of reported scores. This could be due to one

of many reasons: different BM25 implementations, number of passages reranked, and monoBERT training hyperparameters, to name a few.

The third block of the table, rows (1–4), shows the scores when our cross-encoder is trained on BM25-sourced hard negatives and reranks the BM25 runfile. The fourth block, rows (5–8), shows the scores when the cross-encoder is trained on TCT-ColBERTv2-sourced hard negatives and reranks the TCT-ColBERTv2 runfile.

Gao et al. [13] only compare the cross entropy loss to LCE loss with a group size of 8. We generalize these results by additionally considering a group size of 2 with LCE loss and including hinge loss, which can also be viewed as having a group size of 2 but has a different loss formulation. As negative examples are sampled from the same groups of top-ranked passages, both losses benefit from the "localized" effect and the formulation is the only difference.

The cross entropy loss performs on par with the hinge loss when BM25 is used as the retriever, row (1) vs. (2). However, when using TCT-ColBERTv2 as the retriever, the hinge loss demonstrates improved effectiveness over the cross entropy loss by a slight margin, row (5) vs. (6). We suspect this is due to the pairwise loss making better use of the harder negative examples provided by TCT-ColBERTv2.

Another interesting observation is that monoBERT using LCE significantly outperforms monoBERT using the other two losses when TCT-ColBERTv2 forms the first-stage retriever, even when the group size is 2, which hinge loss uses too, row (5–7). However, LCE and hinge losses perform comparably when using BM25 as the retriever, row (2) vs. (3), and fixing the group size at 2. This indicates that the contrastive loss may itself serve as a better approach to distinguish the relevant passage from the negative ones in the ranking task, compared to the hinge loss. It additionally gains from increasing the group size, rows (3) and (7) vs. rows (4) and (8); this is more carefully examined in Sect. 5.3.

Table 2. MRR@10 and Recall@1K of all combinations of training hard negatives retriever and inference first-stage retriever on the development set of the MS MARCO passage dataset. HN refers to the source of Hard Negatives, i.e., the training retriever. All table entries use LCE with group size 8 (one positive sample with seven negative samples). Superscripts indicate significantly higher results ($p < 0.01$ with paired t-tests) after Bonferroni correction.

	HN	First-stage	MRR@10	R@1K
(a)	BM25	BM25	0.391	0.8573
(b)	TCT-ColBERTv2		0.389	
(c)	BM25	TCT-ColBERTv2	0.402^{ab}	0.9690
(d)	TCT-ColBERTv2		0.401^{ab}	

5.2 In-distributional Training Example and Hard Negative

Table 2 presents the effectiveness of the reranker when we vary the retriever for preparing training negatives and generating the development runfile. Rows (a) and (d) here correspond to rows (4) and (8) in Table 1, respectively. To obtain row (b), we use the checkpoint of row (d) to rerank the BM25 runfile. Similarly, we use the checkpoint of row (a) to rerank the TCT-ColBERTv2 runfile for row (c).

It is clear that swapping out the BM25 first-stage retriever with the dense retriever, TCT-ColBERTv2, results in significant improvement irrespective of the retriever used to mine hard negatives, rows (a–b) vs. (c–d). This is reasonable as there is a gap of around 11% in Recall@1K, meaning reranking the runfile produced by TCT-ColBERTv2 would more likely pull up the relevant passages.

We, however, observe no improvement in aligning the retriever used for hard-negative mining with that used for first-stage retrieval in evaluation. In our experiments, changing only the retriever for generating the training data does not yield significant differences in the score when we preserve the first-stage retriever to be the same, row (b) vs. (a) and row (c) vs. (d). This does not agree with the original finding of Gao et al. [13], where they find this alignment critical to the best effectiveness in the MS MARCO *document* ranking task. There are several differences in the experiments that could be responsible for this disagreement. The first is the dataset itself. Although both MS MARCO passage and MS MARCO document are from the same *ad hoc* domain, the document length may impact training data quality. Other possible causes include the range from where we sample the hard negatives, the choice of the first-stage retriever, etc. Based on these results, for the rest of the paper, we use TCT-ColBERTv2 as the first-stage retriever during inference.

Table 3. MRR@10 on the development set of the MS MARCO passage dataset across the choice of group size and retriever used to mine hard negatives. All entries use TCT-ColBERTv2 as the first stage, which has a Recall@1K of 0.9690, as seen in Table 1. Superscripts and subscripts indicate significantly higher results ($p < 0.01$ with paired t-tests) after Bonferroni correction. (e.g., $(\cdot)^{a_{2,4,8}}_{b_2}$ indicates the entry is significantly higher than the results in row (a) with group sizes 2, 4, and 8, and the result in row (b) with group size 2.)

	HN	Group Size				
		2	4	8	16	32
(a)	TCT-ColBERTv2	0.393_{b_2}	0.400_{b_2}	0.401_{b_2}	$0.408^{a_{2,4,8}}_{b_{2,4}}$	$0.414^{a_{2,4,8}}_{b_{2,4,8,16}}$
(b)	BM25	0.381	0.397_{b_2}	0.402_{b_2}	0.403_{b_2}	$0.407_{b_{2,4}}$

5.3 LCE Group Size

We now examine the effect of the group size in the LCE loss, denoted by n, on model effectiveness. This has been studied in the original paper [13] with

$n \in \{2, 4, 6, 8\}$, where additional improvement can always be observed when the group size is increased. We explore the effect of group size in the same manner but increase the range to $\{2, 4, 8, 16, 32\}$. Additionally, we vary the retriever used for hard negative mining. We examine the MRR@10 scores across these settings in Table 3.

As noted by Gao et al. [13], we observe that the primary metric improves as the group size increases. We do so for both choices of the retriever used for hard negative mining. We surprisingly find that the metric does not seem to plateau even when the group size increases to 32 (i.e., with 31 negative samples). We did not experiment on larger group sizes due to hardware limitations,[7] but this suggests that there could be further improvements with improved hardware.

Table 2 does not note any improvements aligning the hard negative mining retriever with that used for first-stage retrieval during inference in the case with group size 8. However, we find that there do exist improvements, especially in group sizes of 2 and 32. We leave further investigation of this unusual observation as future work and use the best setting reported for the rest of the paper.

We submitted the test set run, produced from our most effective configuration, to the MS MARCO passage leaderboard.[8] Table 4 reports our scores and the systems with higher scores on the test set (at the time of our work).[9] The table shows that our best results are quite competitive to the current top results, which use ensembles of multiple cross-encoders, rows (a–c, f), or a multi-stage reranking pipeline, row (e).[10]

Table 4. MRR@10 on the official MS MARCO passage leaderboard.

	Method	Dev MRR@10	Eval MRR@10
(a)	coCondenser [10]	0.443	0.428
(b)	C-COIL + RoBERTa [12]	0.443	0.427
(c)	RocketQA + ERNIE [41]	0.439	0.426
(d)	DR-BERT	0.420	0.419
(e)	expando-mono-duo-T5 [40]	0.420	0.408
(f)	DeepCT + TF-Ranking Ensemble [16]	0.420	0.408
(g)	monoELECTRA	0.414	0.404

[7] The experiment involving a group size of 32 requires 4 Quadro RTX 8000 GPUs (48G memory each) to train with a batch size of 16.

[8] https://microsoft.github.io/MSMARCO-Passage-Ranking-Submissions/leaderboard.

[9] We copy the best results from each group and discard anonymous results.

[10] We cannot compare with the DR-BERT system, as we do not find its resources publicly available online.

6 Conclusion

In this paper, we replicate the LCE loss proposed by Gao et al. [13] on a different codebase and generalize their findings to the MS MARCO passage dataset. We confirm the superiority of LCE loss to the cross entropy and hinge loss on the passage ranking task, with improved effectiveness when using a better first-stage retrieval method like TCT-ColBERTv2 during inference. However, we argue that more exploration is necessary to conclude if the alignment between the training and inference first-stage retriever is essential across group sizes. Finally, we confirm that the effectiveness can be further strengthened by increasing the number of hard negatives in each group.

Acknowledgments. This research was supported in part by the Canada First Research Excellence Fund and the Natural Sciences and Engineering Research Council (NSERC) of Canada. Computational resources were provided by Compute Ontario and Compute Canada.

References

1. Akkalyoncu Yilmaz, Z., Yang, W., Zhang, H., Lin, J.: Cross-domain modeling of sentence-level evidence for document retrieval. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3490–3496, November 2019
2. Bajaj, P., et al.: MS MARCO: a human generated machine reading comprehension dataset. arXiv preprint arXiv:1611.09268v3 (2018)
3. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: ELECTRA: pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555 (2020)
4. Dai, Z., Callan, J.: Deeper text understanding for IR with contextual neural language modeling. In: Proceedings of the 42nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019), pp. 985–988 (2019)
5. Dai, Z., Callan, J.: Context-aware document term weighting for ad-hoc search. In: Proceedings of The Web Conference 2020, p. 1897–1907 (2020)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186 (2019)
7. Dietz, L., Verma, M., Radlinski, F., Craswell, N.: TREC complex answer retrieval overview. In: Proceedings of the Twenty-Seventh Text REtrieval Conference (TREC 2018) (2018)
8. Formal, T., Piwowarski, B., Clinchant, S.: SPLADE: sparse lexical and expansion model for first stage ranking. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021), pp. 2288–2292 (2021)

9. Gao, L., Callan, J.: Condenser: a pre-training architecture for dense retrieval. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 981–993, November 2021
10. Gao, L., Callan, J.: Unsupervised corpus aware language model pre-training for dense passage retrieval. arXiv preprint arXiv:2108.05540 (2021)
11. Gao, L., Dai, Z., Callan, J.: Understanding BERT rankers under distillation. In: Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval. ICTIR 2020, pp. 149–152 (2020)
12. Gao, L., Dai, Z., Callan, J.: COIL: Revisit exact lexical match in information retrieval with contextualized inverted list. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 3030–3042, June 2021
13. Gao, L., Dai, Z., Callan, J.: Rethink training of BERT rerankers in multi-stage retrieval pipeline. In: Hiemstra, D., Moens, M.-F., Mothe, J., Perego, R., Potthast, M., Sebastiani, F. (eds.) ECIR 2021. LNCS, vol. 12657, pp. 280–286. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72240-1_26
14. Guo, J., Fan, Y., Ai, Q., Croft, W.B.: A deep relevance matching model for ad-hoc retrieval. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM 2016, pp. 55–64 (2016)
15. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 9, pp. 297–304 (2010)
16. Han, S., Wang, X., Bendersky, M., Najork, M.: Learning-to-rank with BERT in TF-ranking. arXiv preprint arXiv:2004.08476 (2020)
17. Hofstätter, S., Lin, S.C., Yang, J.H., Lin, J., Hanbury, A.: Efficiently teaching an effective dense retriever with balanced topic aware sampling. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021), SIGIR 2021, pp. 113–122 (2021)
18. Hofstätter, S., Zlabinger, M., Hanbury, A.: Interpretable & time-budget-constrained contextualization for re-ranking. In: Proceedings of the 24th European Conference on Artificial Intelligence (ECAI 2020), Santiago de Compostela, Spain, pp. 513–520 (2020)
19. Hui, K., Yates, A., Berberich, K., de Melo, G.: PACRR: a position-aware neural IR model for relevance matching. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 1049–1058 (2017)
20. Jiang, K., Pradeep, R., Lin, J.: Exploring listwise evidence reasoning with T5 for fact verification. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp. 402–410 (2021)
21. Karpukhin, V., et al.: Dense passage retrieval for open-domain question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6769–6781 (2020)
22. Khattab, O., Zaharia, M.: ColBERT: efficient and effective passage search via contextualized late interaction over BERT. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 39–48 (2020)
23. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

24. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: a lite BERT for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942 (2019)
25. Li, C., Yates, A., MacAvaney, S., He, B., Sun, Y.: PARADE: passage representation aggregation for document reranking. arXiv preprint arXiv:2008.09093 (2020)
26. Lin, J., Ma, X., Lin, S.C., Yang, J.H., Pradeep, R., Nogueira, R.: Pyserini: a Python toolkit for reproducible information retrieval research with sparse and dense representations. In: Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021), pp. 2356–2362 (2021)
27. Lin, J., Nogueira, R., Yates, A.: Pretrained transformers for text ranking: BERT and beyond. arXiv preprint arXiv:2010.06467 (2020)
28. Lin, S.C., Yang, J.H., Lin, J.: Distilling dense representations for ranking using tightly-coupled teachers. arXiv:2010.11386 (2020)
29. Lin, S.C., Yang, J.H., Lin, J.: In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval. In: Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021), pp. 163–173 (2021)
30. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
31. Ma, X., Guo, J., Zhang, R., Fan, Y., Ji, X., Cheng, X.: B-PROP: bootstrapped pre-training with representative words prediction for ad-hoc retrieval. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021) (2021)
32. Ma, X., Guo, J., Zhang, R., Fan, Y., Ji, X., Cheng, X.: PROP: pre-training with representative words prediction for ad-hoc retrieval. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining (2021)
33. MacAvaney, S., Yates, A., Cohan, A., Goharian, N.: CEDR: contextualized embeddings for document ranking. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1101–1104 (2019)
34. Nogueira, R., Cho, K.: Passage re-ranking with BERT. arXiv preprint arXiv:1901.04085 (2019)
35. Nogueira, R., Jiang, Z., Pradeep, R., Lin, J.: Document ranking with a pretrained sequence-to-sequence model. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 708–718 (2020)
36. Nogueira, R., Yang, W., Cho, K., Lin, J.: Multi-stage document ranking with BERT. arXiv preprint arXiv:1910.14424 (2019)
37. Pradeep, R., Ma, X., Nogueira, R., Lin, J.: Scientific claim verification with VerT5erini. In: Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis, pp. 94–103 (2021)
38. Pradeep, R., Ma, X., Nogueira, R., Lin, J.: Vera: Prediction techniques for reducing harmful misinformation in consumer health search. In: Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021) (2021)
39. Pradeep, R., Ma, X., Zhang, X., Cui, H., Xu, R., Nogueira, R., Lin, J.: H_2oloo at TREC 2020: when all you got is a hammer... deep learning, health misinformation, and precision medicine. In: Proceedings of the Twenty-Ninth Text REtrieval Conference (TREC 2020) (2020)
40. Pradeep, R., Nogueira, R., Lin, J.: The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. arXiv preprint arXiv:2101.05667 (2021)

41. Qu, Y., et al.: RocketQA: an optimized training approach to dense passage retrieval for open-domain question answering. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 5835–5847 (2021)
42. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(140), 1–67 (2020)
43. Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., Gurevych, I.: BEIR: a heterogenous benchmark for zero-shot evaluation of information retrieval models. arXiv preprint arXiv:2104.08663, April 2021
44. Wolf, T., et al.: HuggingFace's transformers: state-of-the-art natural language processing. arXiv preprint arXiv:1910.03771 (2019)
45. Xiong, C., Dai, Z., Callan, J., Liu, Z., Power, R.: End-to-end neural ad-hoc ranking with kernel pooling. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2017, pp. 55–64 (2017)
46. Xiong, L., et al.: Approximate nearest neighbor negative contrastive learning for dense text retrieval. In: Proceedings of the 9th International Conference on Learning Representations (ICLR 2021) (2021)
47. Yang, P., Fang, H., Lin, J.: Anserini: enabling the use of Lucene for information retrieval research. In: Proceedings of the 40th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017), pp. 1253–1256 (2017)
48. Yang, W., et al.: End-to-end open-domain question answering with BERTserini. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), pp. 72–77 (2019)
49. Yates, A., Arora, S., Zhang, X., Yang, W., Jose, K.M., Lin, J.: Capreolus: a toolkit for end-to-end neural ad hoc retrieval. In: Proceedings of the 13th International Conference on Web Search and Data Mining, pp. 861–864 (2020)
50. Yates, A., Jose, K.M., Zhang, X., Lin, J.: Flexible IR pipelines with Capreolus. In: Proceedings of the 29th International Conference on Information and Knowledge Management (CIKM 2020) (2020)
51. Zhan, J., Mao, J., Liu, Y., Guo, J., Zhang, M., Ma, S.: Optimizing dense retrieval model training with hard negatives. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021), pp. 1503–1512 (2021)
52. Zhang, E., et al.: Covidex: neural ranking models and keyword search infrastructure for the COVID-19 open research dataset. In: Proceedings of the First Workshop on Scholarly Document Processing, pp. 31–41 (2020)
53. Zhang, X., Yates, A., Lin, J.: Comparing score aggregation approaches for document retrieval with pretrained transformers. In: Hiemstra, D., Moens, M.-F., Mothe, J., Perego, R., Potthast, M., Sebastiani, F. (eds.) ECIR 2021, Part II. LNCS, vol. 12657, pp. 150–163. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72240-1_11