Distillation Methods for Cross-Encoder-Based Reranking

Luke Tchang
Stanford University
ltchang@stanford.edu

Abstract

Reranking models are critical parts of production-ready semantic retrieval pipelines. Many multi-stage pipelines start with dense embedding models or sparse lexical indices for first-stage retrieval, then follow up with a second-stage reranking model. Because neural rerankers perform full attention on both the query and document, they are often able to achieve strong accuracy gains over the first-stage rankings. Still, due to their high compute cost, the latency overhead of most neural rerankers is noticeable—on the order of several hundred milliseconds to a couple of seconds. In this paper, we explore a combination of methods for distilling large reranking models including student-teacher distillation, curriculum learning, and hard negative sampling. We train several deberta-v3-large rerankers, with our top model outperforming the prior state-of-the-art reranking model in the 400M parameter range by 0.77% and 1.04% on two of the three BEIR Q&A benchmarks, respectively. We find that combining distillation loss with direct contrastive loss has a strong additive benefit that outperforms either the distillation or contrastive methods alone. In addition, in our ablation studies, we find the relative benefits of distillation increase the larger the model size gap between the teacher and student.

1 Introduction

Retrieval is a core component of many products today that have some element of search, question-answering, or recommendation. As large language models (LLMs) and retrieval-augmented generation (RAG) [1] have become more popular, the sophistication and demands of information retrieval systems have increased.

Most retrieval pipelines begin with a combination of sparse lexical search and dense vector search—first-stage retrieval. Dense embedding models take a query or document and output a fixed-dimension vector that captures the meaning of the text. When searching for relevant documents, you perform an approximate nearest-neighbor (ANN) search, finding the document vectors with the highest inner product or cosine similarity to the query vector. This is efficient because documents can be embedded and indexed offline—only the query needs to be embedded at query time.

Second-stage retrieval aims to improve the ordering of the first-stage candidates. Most reranking models are cross-encoder Transformer models [2], which take both the query and document and output a relevance score. Because the reranking model performs full self-attention on both the query and document, more complex relationships can be captured, leading to better ranking performance. Still, because attention is computed online at least once per query-document pair, the latency of second-stage retrieval is often substantial.

In this paper, we explore several methods for distilling reranking models to improve their efficiency. The main contributions of this paper are:

- We train a state-of-the-art small reranking model distilled from NV-RerankQA-Mistral-4B-v3.

- We provide a comprehensive study of methods for reranker distillation using various loss functions, hard negative sampling, and curriculum learning.
- We present ablation studies investigating the effectiveness of curriculum learning and distillation across various student model sizes.

2 Related Work

2.1 LLM-Based Ranking

Recently, there has been a large volume of work on LLM-based ranking. Sun et al. presented RankGPT [3], a ranking system that uses an LLM to list-wise rerank a set of candidate documents in a sliding window fashion. They achieved state-of-the-art results when reranking the top 100 BM25 candidates of DL19 and DL20 with GPT-4. We initially considered distilling models from GPT-4 but found that general-purpose, generative language models were unable to effectively handle hard negatives surfaced by strong embedding models, often worsening ranking performance compared to the embedding model alone.

2.2 Contrastive Training For Reranking Models

Given the substantial quantity of large datasets with sparse relevance labels, contrastive learning methods have gained popularity for training both embedding and reranking models. Given a pair of positive and negative documents, the goal is to maximize the distance between the two documents' vector representations or relevance scores. Moreira et al. presented NV-RerankQA-Mistral-4B-v3 [4], a state-of-the-art reranking model fine-tuned from Mistral 7B using a contrastive loss function and pruned down to 4B parameters. In addition, they also trained a deberta-v3-large model using the same methods and achieved strong results.

2.3 Margin-Based Distillation

Distillation methods for cross-encoders have previously been explored for BERT-style models. Hofstätter et al. [7] presented a novel distillation method that minimizes the difference between the positive-negative distance of the teacher predictions and the positive-negative distance of the student predictions. At the time, their distilled BERT-based model achieved state-of-the-art performance on the DL19 and DL20 benchmarks. However, given the model was not trained on hard negative samples (i.e. it used the top 100 BM25 candidates), it performed poorly on hard negatives surfaced by embedding models in the NV-RerankQA paper [4].

3 Approach

We take the current state-of-the-art 4B parameter reranking model, NV-RerankQA-Mistral-4B-v3 [4], and use it as a teacher model to train smaller reranking models, namely deberta-v3-large and deberta-v3-base which are 405M and 184M parameters, respectively [5]. We make use of hard negative sampling in the training data pipeline, a combination of margin-based distillation and contrastive loss functions during training, and curriculum learning for the training run stages. We provide details on our methods below.

3.1 Data Pipeline

For training data, we use train splits from the BEIR question-answering datasets (NQ, HotpotQA, and FiQA). First, we perform hard negative sampling for the initial retrieval of training samples using the embedding model NV-EmbedQA-Mistral-7B-v2 [6]. Then, we rerank all training samples using the teacher reranking model and apply the *TopK-PercPos* heuristic [6] to filter out potential false negatives.

3.2 Training Methods

We compare and combine two loss functions: InfoNCE (Info Noise-Contrastive Estimation) and MarginMSE (Margin Mean-Squared-Error). InfoNCE is a contrastive loss that maximizes the distance

between the predictions of positive and negative pairs. It is used in NV-RerankQA-Mistral-4B-v3 [4] and shown in Equation 1, where ϕ denotes the output score of the reranker model. MarginMSE, a variant of MSE-based distillation loss, minimizes the difference between the positive-negative distance of the teacher predictions and the positive-negative distance of the student predictions [7]. MarginMSE is outlined in Equation 2 with ϕ_t and ϕ_s denoting the outputs of the teacher and student models, respectively. Drawing on the early work of Hinton et al.[8], we also combine both distillation and contrastive loss functions, effectively aiming to "ground" the distillation-based learning with information from the original objective. The combined loss function is shown in Equation 3 and, as shown in our experiments section, produces the strongest results of the three losses.

$$L_{InfoNCE} = -\log \frac{\exp(\phi(q, d^+)/\tau)}{\exp(\phi(q, d^+)/\tau) + \sum_{i=1}^{N} \exp(\phi(q, d_i^-)/\tau)} \tag{1}$$

$$L_{MarginMSE} = \text{MSE}(\phi_s(q, d^+) - \phi_s(q, d^-), \phi_t(q, d^+)) - \phi_t(q, d^-)) \tag{2}$$

$$L_{Combined} = \alpha L_{MarginMSE} + \beta L_{InfoNCE} \tag{3}$$

Additionally, we employ curriculum learning to increase training difficulty over time, inspired by Zeng, et al [9]. For the first 50% of training samples, we use the top 100 reranked candidates. For the next two 25% splits, we use the top 50 and top 20 candidates, respectively, progressively increasing the concentration of difficult hard negatives.

3.3 Model Architecture

We use the DeBERTa V3 family of models, which are encoder-only pre-trained Transformer models with bidirectional attention [5]. We configure our cross-encoder models to use mean pooling to

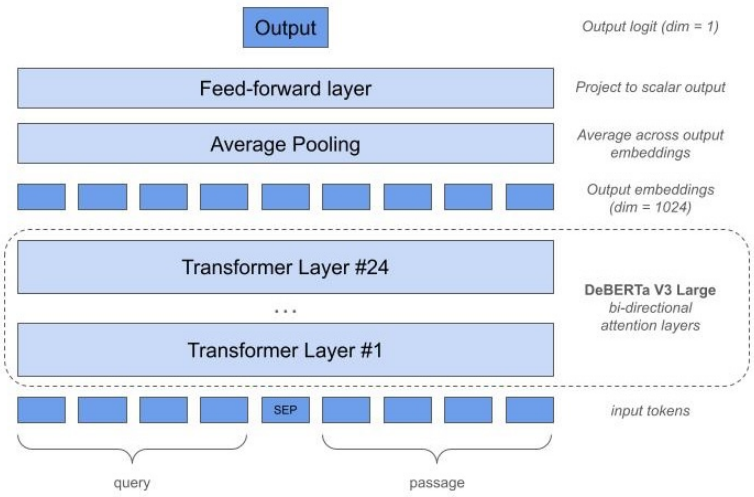

Figure 1: Architecture of deberta-v3-large reranker

combine information across the final hidden states. The mean pooled output vector is then passed through a fully-connected layer to project the hidden output to a scalar relevance score. Our architecture is illustrated in Figure 1.

4 Experiments

4.1 Evaluation Setup

We evaluate our model on the three question-answering datasets from the BEIR benchmark (NQ, FiQA, and HotpotQA) [10], the same datasets the teacher reranking model was evaluated on. We first use NV-EmbedQA-Mistral7B-v2 as the first-stage retriever to gather the top 100 documents, then rerank them using our various reranking models. Our primary metric is NDCG@10.

4.2 Training Setup

For our main experiments, we train and evaluate several models using the various loss functions described in Section 3.2. For our combined loss training runs, we use $\alpha = 0.7$ and $\beta = 0.3$, following the pattern of more heavily weighing distillation loss than the direct objective loss from Hinton et al. [8]. All training runs use data surfaced from hard negative sampling and false positive filtering and follow the curriculum learning schedule in Section 3.2. We train our models for a maximum of 12 hours on a set of 2x H100 GPUs using bf16 mixed precision.

4.3 Benchmark Results

As shown in our benchmarks in Table 1, combined distillation and contrastive loss overall outperform the prior state-of-the-art deberta-v3-large model from the NV-RerankQA paper. More specifically, it improves NDCG@10 by 1.04% and 0.77% on the NQ and HotpotQA datasets, respectively. In addition, MarginMSE also outperforms both our InfoNCE model and Nvidia's deberta-v3-large model on the NQ and HotpotQA datasets.

Model	Loss	NQ	HotpotQA	FiQA	Average (All)
***Embedding*: NV-EmbedQA-Mistral7B-v2**	InfoNCE	0.7216	0.8109	0.6194	0.7173
Teacher Model					
+ `nv-rerankqa-mistral-4b-v3`	InfoNCE	0.7717	0.8857	0.6152	0.7575
Small Models					
+ `deberta-v3-large (nv-rerank)`	InfoNCE	0.7486	0.8700	**0.6055**	0.7412
+ `deberta-v3-large (ours)`	Combined	**0.7559**	**0.8767**	0.5922	**0.7416**
+ `deberta-v3-large (ours)`	MarginMSE	0.7503	0.8720	0.5769	0.7330
+ `deberta-v3-large (ours)`	InfoNCE	0.7452	0.8665	0.5760	0.7292

Table 1: NDCG@10 for the deberta-v3-large models on the BEIR Q&A datasets, all trained with curriculum learning

Despite our best model having a better average than the Nvidia deberta-v3-large model, we see weaker results than expected on FiQA. We hypothesize that this is due to the fact that the teacher model scores worse than the embedding model alone on FiQA specifically, causing adverse false positive filtering and potentially worsened distillation results.

5 Ablation Studies and Analysis

In this section, we present two ablation studies and an analysis of inference latency for different model sizes. The first ablation study compares the results of models trained with and without curriculum training. The second ablation study investigates how the effectiveness of distillation-based training changes as the student-teacher model size gap increases.

5.1 Curriculum Learning

To isolate the effects of curriculum learning, we train each of our models with and without curriculum learning phases. The results are shown in Table 2. We see that curriculum learning consistently

Model	Loss	Curriculum	NQ	HotpotQA	FiQA	Average (All)
***Embedding*: NV-EmbedQA-Mistral7B-v2**	InfoNCE	N/A	0.7216	0.8109	0.6194	0.7173
`+ deberta-v3-large (ours)`	Combined	Yes	**0.7559**	**0.8767**	**0.5922**	**0.7416**
`+ deberta-v3-large (ours)`	Combined	No	0.7554	0.8759	0.5913	0.7408
`+ deberta-v3-large (ours)`	MarginMSE	Yes	0.7503	0.8720	0.5769	0.7330
`+ deberta-v3-large (ours)`	MarginMSE	No	0.7472	0.8718	0.5721	0.7304
`+ deberta-v3-large (ours)`	InfoNCE	Yes	0.7452	0.8665	0.5760	0.7292
`+ deberta-v3-large (ours)`	InfoNCE	No	0.7435	0.8662	0.5749	0.7282

Table 2: NDCG@10 for the deberta-v3-large models trained with and without curriculum learning

improves average performance for models across the board by anywhere from 0.1% to 0.36%. During training, this is also noticeable, as the overall training loss decreases to lower levels when using curriculum learning than when just performing the whole training run without increasing difficult hard negative concentration. Thus curriculum learning proved a consistent method for slight additional performance gains.

5.2 Model Size

Based on the results in Table 3, we surprisingly find that the relative effectiveness of distillation methods increases the wider the model size gap is between the student and teacher.

Model	Loss	NQ	HotpotQA	FiQA	Average (All)
***Embedding*: NV-EmbedQA-Mistral7B-v2**	InfoNCE	0.7216	0.8109	0.6194	0.7173
`+ deberta-v3-base (ours)`	Combined	**0.7157**	**0.8648**	0.5092	**0.6965**
`+ deberta-v3-base (ours)`	MarginMSE	0.6966	0.8588	**0.5271**	0.6935
`+ deberta-v3-base (ours)`	InfoNCE	0.6620	0.8468	0.4999	0.6696

Table 3: NDCG@10 for the deberta-v3-base models on the BEIR Q&A datasets

The table shows that MarginMSE provides only a 0.52% performance improvement over InfoNCE when training deberta-v3-large, a 405M parameter model. However, when training deberta-v3-base, which has only 184M parameters, MarginMSE provides a 3.67% relative improvement over the same model trained using InfoNCE. We can see this more clearly in Figure 2, where the performance gap between InfoNCE and the loss functions that use MarginMSE is wider for deberta-v3-base but shrinks for deberta-v3-large.

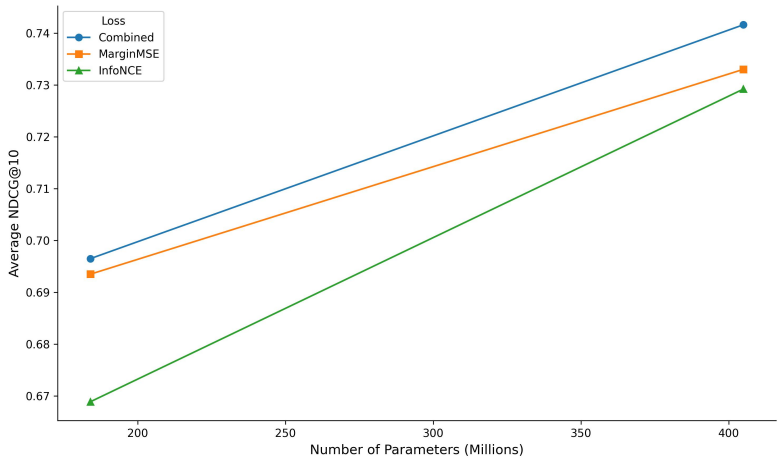

Figure 2: Average NDCG@10 plotted against model size for various loss functions. `mi`, `dl`, `db` denote Mistral-4B, deberta-v3-large, and deberta-v3-base, respectively.

This suggests that the smaller ranking models benefit more from having a strong teacher model to learn from and that the additional information provided by distillation can aid smaller models in learning otherwise more difficult objectives.

5.3 Latency

We also measure and analyze the average reranking latency of the teacher model as well as our deberta-v3-large and deberta-v3-base models against two of the three BEIR Q&A datasets. Each evaluation is carried out on a single H100 at half-precision.

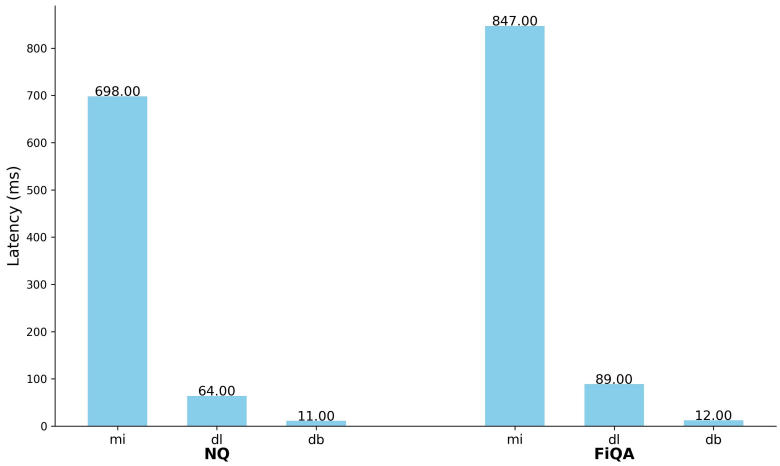

Figure 3: Latency of reranking top 100 candidates for different models. `mi` indicates Mistral-4B, `dl` indicates deberta-v3-large, and `db` indicates deberta-v3-base.

The NQ dataset is on average 123 tokens per passage while FiQA passages are roughly 50% longer at 191 tokens. As shown in Figure 3, deberta-v3-large is roughly 10x faster than the Mistral-4B model for both input passage lengths. While deberta-v3-base is roughly 2.2x smaller than deberta-v3-large, its inference latency is 6-8x faster. We hypothesize that the non-linear speedup from deberta-v3-large to deberta-v3-base is due to deberta-v3-base having half the number of Transformer layers of deberta-v3-large (12 vs 24 layers). In contrast, deberta-v3-large has fewer parameters than the Mistral-4B model but has more Transformer layers (24 vs 16 layers). Most importantly, however, the measurements show that the deberta-v3 models fall into a much more usable latency range—under 100 milliseconds. For many interactive products that require sub-200 millisecond search latency, this improvement is substantial.

6 Conclusion

In this paper, we provide a comprehensive study of methods for training and distilling small reranking models. We demonstrate that grounding distillation-based loss with an additional contrastive loss term yields superior results to either loss alone, resulting in a model that outperforms the prior state-of-the-art reranking model in the 400M parameter range.

In addition, we describe our model architecture, data pipeline, and training methods for adapting and fine-tuning deberta-v3-large as a cross-encoder, summarizing our benchmark results on the BEIR Q&A datasets. Through our ablation studies, we find that curriculum learning consistently yields slight performance gains and that distillation methods are particularly effective when training hyper-efficient and compact reranking models. Lastly, we present an analysis of reranking latency across models of size 4B, 405M, and 184M parameters, making clearer the relative tradeoff between latency and accuracy.

References

[1] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, Douwe Kiela. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.* arXiv preprint arXiv:2005.11401, 2020. https://arxiv.org/abs/2005.11401.

[2] Rodrigo Nogueira, Kyunghyun Cho. *Passage Re-ranking with BERT.* arXiv preprint arXiv:1901.04085, 2019. https://arxiv.org/abs/1901.04085.

[3] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, Zhaochun Ren. *Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents.* arXiv preprint arXiv:2304.09542, 2023. https://arxiv.org/abs/2304.09542.

[4] Gabriel de Souza P. Moreira, Ronay Ak, Benedikt Schifferer, Mengyao Xu, Radek Osmulski, Even Oldridge. *Enhancing Q&A Text Retrieval with Ranking Models: Benchmarking, fine-tuning and deploying Rerankers for RAG,* arXiv preprint arXiv:2409.07691, 2024. https://arxiv.org/abs/2409.07691.

[5] Pengcheng He, Jianfeng Gao, Weizhu Chen. *DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing.* arXiv preprint arXiv:2111.09543, 2021. https://arxiv.org/abs/2111.09543.

[6] Gabriel de Souza P Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. 2024. NV-Retriever: Improving text embedding models with effective hard-negative mining. arXiv preprint arXiv:2407.15831 (2024) https://arxiv.org/pdf/2405.17428

[7] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, Allan Hanbury. *Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation,* arXiv preprint arXiv:2010.02666, 2020. https://arxiv.org/abs/2010.02666.

[8] Geoffrey Hinton, Oriol Vinyals, Jeff Dean. *Distilling the Knowledge in a Neural Network.* arXiv preprint arXiv:1503.02531, 2015. https://arxiv.org/abs/1503.02531.

[9] Hansi Zeng, Hamed Zamani, Vishwa Vinay. *Curriculum Learning for Dense Retrieval Distillation.* arXiv preprint arXiv:2204.13679, 2022. https://arxiv.org/abs/2204.13679.

[10] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, Iryna Gurevych. *BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models.* arXiv preprint arXiv:2104.08663, 2021. https://arxiv.org/abs/2104.08663.