Rank-DistiLLM: Closing the Effectiveness Gap Between Cross-Encoders and LLMs for Passage Re-Ranking

Ferdinand Schlatt[1], Maik Fröbe[1], Harrisen Scells[2,6],
Shengyao Zhuang[3,4], Bevan Koopman[3], Guido Zuccon[4],
Benno Stein[5], Martin Potthast[2,6,7], and Matthias Hagen[1]

[1] Friedrich-Schiller-Universität Jena [2] University of Kassel [3] CSIRO
[4] University of Queensland [5] Bauhaus-Universität Weimar
[6] hessian.AI [7] ScadDS.AI

Abstract. Cross-encoders distilled from large language models (LLMs) are often more effective re-rankers than cross-encoders fine-tuned on manually labeled data. However, distilled models do not match the effectiveness of their teacher LLMs. We hypothesize that this effectiveness gap is due to the fact that previous work has not applied the best-suited methods for fine-tuning cross-encoders on manually labeled data (e.g., hard-negative sampling, deep sampling, and listwise loss functions). To close this gap, we create a new dataset, Rank-DistiLLM. Cross-encoders trained on Rank-DistiLLM achieve the effectiveness of LLMs while being up to 173 times faster and 24 times more memory efficient. Our code and data is available at https://github.com/webis-de/ECIR-25.

1 Introduction

Cross-encoders [1, 48, 72] are among the most effective passage re-rankers [38, 59], but require large amounts of labeled data for fine-tuning. In contrast, large language models (LLMs) require no further fine-tuning and are often more effective than cross-encoders [53, 54, 62]. The main drawback of using LLMs is their computational cost, making them impractical for production search engines. However, LLMs can be used to create training data for fine-tuning cross-encoders.

Previous work [3, 63] has shown that cross-encoders distilled from LLMs are more effective than cross-encoders fine-tuned on manually labeled data. But the distilled cross-encoders do not reach the effectiveness of their teacher LLMs because many of the best practices for fine-tuning on manually labeled data were not considered: no "hard-negative" sampling was used [33, 52], the distillation rankings were not deep enough [73], and no listwise losses were used [33].

In this paper, we close the effectiveness gap between distilled cross-encoders and their teacher LLMs. We analyze the impact of the first-stage retrieval model and the ranking depth on distilled cross-encoders by introducing a new dataset, Rank-DistiLLM, and propose a novel listwise loss function for distillation. As a result, we provide a new distillation method for cross-encoders that is as effective as state-of-the-art ranking LLMs while being orders of magnitude more efficient.

2 Related Work

MS MARCO is the most commonly used dataset for fine-tuning cross-encoders as it contains over 500k query–passage pairs [47]. However, most queries only have a single labeled passage. This label sparsity has two implications: (1) the options for suitable loss functions are limited, and (2) "non-relevant" passages must be sampled heuristically.

Regarding the first implication, listwise losses produce the most effective models [33, 52, 73]. They use a single relevant passage and a set of k heuristically sampled "non-relevant" passages. Generally, a higher k produces more effective models—with $k = 36$ being the highest reported value [73]. We rely on recent work on memory-efficient fused-attention kernels [27, 28, 41] to fine-tune models on up to $k = 100$ passages. Regarding the second implication, "hard negative" sampling, i.e., using an effective first-stage retrieval model to sample "non-relevant" samples, has produced the most effective models [33, 52, 73]. For instance, models fine-tuned on negatives sampled from ColBERTv2 [60] are more effective than those fine-tuned on negatives sampled from BM25 [58]. However, MS MARCO contains passages that are more relevant than the labeled passage [2], leading to noisy training data.

To obtain less noisy data, Sun et al. [62] proposed fine-tuning a cross-encoder on the rankings generated by an LLM applied in zero-shot manner. Models distilled from this dataset are more effective in select scenarios than those fine-tuned on MS MARCO. More recently, Baldelli et al. [3] released a dataset that yields more effective cross-encoders by providing more samples per query and using a mixture of first-stage retrieval models. However, an effectiveness gap between a cross-encoder and its teacher LLMs remains. We investigate if this gap can be closed by applying the insights mentioned above to LLM distillation.

3 Improving the Fine-tuning of Cross-Encoders

Preliminaries. A cross-encoder outputs contextualized embeddings for every token of a given input sequence [CLS] q [SEP] d [SEP], where q and d are sequences of query and passage tokens [29]. A linear layer is then applied to the [CLS] token's contextualized embedding to compute the relevance score s_d.

3.1 Traditional Fine-Tuning on MS MARCO

Loss When fine-tuning cross-encoders on data sampled from MS MARCO, previous work obtains the most effective models by using the listwise InfoNCE loss [50] (also called listwise softmax cross-entropy [7, 73] or localized contrastive estimation (LCE) [33, 52]). Given a set of passages $\mathcal{D}$, of which one is relevant, $d^+ \in \mathcal{D}$, InfoNCE is defined as:

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp(s_{d^+})}{\sum_{d \in \mathcal{D}} \exp(s_d)}.$$

Data To obtain the highest possible effectiveness, $\mathcal{D}$ should be as large as possible. Additionally, the best available first-stage retrieval model should retrieve the other passages $\mathcal{D}^- = \mathcal{D} \setminus \{d^+\}$. Following Pradeep et al. [52], we retrieve the top 200 passages for all MS MARCO training queries with ColBERTv2 [60] and then randomly sample 7 hard-negatives.

3.2 Improved Fine-Tuning on LLM Distillation Data

Loss Instead of a set of passages $\mathcal{D}$, LLM distillation data consists of a list of passages $\mathcal{R} = (d_1, d_2, \dots, d_n)$ for a query q ranked by an LLM. Previous work [3, 62] uses the pairwise RankNet loss function [8] for fine-tuning:

$$\mathcal{L}_{\text{RankNet}} = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \log(1 + \exp(s_{d_j} - s_{d_i})).$$

To test if listwise loss functions also improve the effectiveness of cross-encoders distilled from LLM rankings, we propose a new loss function based on the Approx family of loss functions [55]. Approx loss functions compute a smooth approximation $\hat{\pi}(d)$ of a passage's rank based on all passages' scores. Our new loss, Approx Discounted Rank MSE (ADR-MSE), computes the mean squared error between a passage's actual and approximated rank. Inspired by nDCG, we also apply a logarithmic discount to give higher-ranked passages a higher weight:

$$\mathcal{L}_{\text{ADR-MSE}} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\log_2(i+1)} (i - \hat{\pi}(d_i))^2.$$

Data To our knowledge, only two datasets for distilling cross-encoders from LLMs have been released. Sun et al. [62] released the first dataset (RankGPT) consisting of the top 20 passages retrieved by BM25 [58] and re-ranked by RankGPT-3.5 for 100k queries from MS MARCO. Baldelli et al. [3] released another dataset (TWOLAR) of the top 30 passages retrieved by three different retrieval models (BM25, DRAGON [42], and SPLADE [31]) and re-ranked by RankGPT-3.5 for a total of 20k-queries from MS MARCO. TWOLAR produces more effective models, but whether the improved first-stage retrieval models, deeper rankings, or both in combination lead to better effectiveness is unclear.

We create Rank-DistiLLM to systematically investigate the effect of the first-stage retrieval model and the rank depth on a cross-encoder's downstream effectiveness. We retrieve the top 100 passages using BM25 and ColBERTv2 for 10k randomly sampled queries from the MS MARCO training set. We then use RankZephyr [54], an open-source alternative to RankGPT, to re-rank them. RankZephyr was fine-tuned using the original RankGPT distillation data and additional higher-quality RankGPT-4 distillation data. We refer to this dataset as RankGPT+. To evaluate the effect of ranking depth, we subsample additional datasets by removing all passages that were not within the top 10, 25, and 50 passages of the first-stage retrieval. We release Rank-DistiLLM to the community to facilitate further research.[1]

[1] https://zenodo.org/records/12528410

4 Evaluation

Experimental Setup We mostly follow Pradeep et al. [52] for fine-tuning cross-encoders. We use HuggingFace [71] $\text{ELECTRA}_{\text{BASE}}$ or $\text{ELECTRA}_{\text{LARGE}}$ [10] checkpoints as starting points.[2,3] For fine-tuning using MS MARCO [47] labels, we data described in Section 3.1 and fine-tune for 20k steps using InfoNCE loss. For fine-tuning on LLM distillation data, we compare our new Rank-DistiLLM datasets with the previously released datasets described in Section 3.2. We use the TREC Deep Learning 2021 and 2022 tracks [24, 25] as validation sets and fine-tune until nDCG@10 does not improve for 100 steps using either RankNet [8] or our novel ADR-MSE loss (using $\alpha = 1$). All models are fine-tuned using a batch size of 32 and the AdamW [43] optimizer with a 10^{-5} learning rate. We truncate queries longer than 32 tokens and passages longer than 256 tokens. All models are trained on a single NVIDIA A100 40GB GPU. We used the following packages and frameworks to implement models and run experiments: ir_datasets, ir-measures, Jupyter, Lightning, Lightning IR, matplotlib, NumPy, pandas, PyTerrier, PyTorch, and SciPy [30, 34, 39, 40, 44–46, 51, 61, 64, 65].

4.1 In-Domain Effectiveness

Table 1 lists nDCG@10 scores of monoELECTRA, a cross-encoder using ELECTRA [10] as the backbone encoder, fine-tuned on the data mentioned in Section 3.2, and evaluated on the TREC DL 2019 and 2020 tasks when re-ranking the top 100 passages retrieved by BM25 and ColBERTv2. We compare our model with RankGPT-4, RankZephyr, $\text{monoT5}_{\text{3B}}$ [49], $\text{RankT5}_{\text{3B}}$ [73], and monoELECTRA fine-tuned using MS MARCO labels. We use a t-test to compute the significance of differences of all models to the best monoELECTRA model fine-tuned using our Rank-DistiLLM dataset with ($p < 0.05$, Holm-Bonferroni-corrected).

Labeled Data vs LLM Distillation Our results align with Baldelli et al. [3] in that a monoELECTRA model only fine-tuned using our ColBERTv2 Rank-DistiLLM dataset is more effective than monoELECTRA fine-tuned using MS MARCO labels. However, the differences are not statistically significant. Also in line with Baldelli et al., we find two-stage fine-tuning, i.e., first fine-tuning on MS MARCO and continuing to fine-tune on distillation data, to be effective. The two-stage fine-tuned models are more effective than their single-stage fine-tuned counterparts in nearly all cases, but the differences are, again, not statistically significant. In summary, LLM distillation improves effectiveness for in-domain re-ranking, but manual judgements with hard-negative mining still produces competitive models.

The $\text{monoT5}_{\text{3B}}$ and $\text{RankT5}_{\text{3B}}$ models demonstrate that larger models can achieve higher effectiveness when fine-tuned on MS MARCO labels. To investigate if larger models also improve effectiveness for our Rank-DistiLLM dataset, we fine-tuned a $\text{monoELECTRA}_{\text{Large}}$ model with the two-stage fine-tuning paradigm.

[2] google/electra-base-discriminator

[3] google/electra-large-discriminator

Table 1: Comparison of nDCG@10 on TREC DL 2019 and 2020 of baseline models with monoELECTRA directly fine-tuned (Single-Stage) or further fine-tuned from an already fine-tuned model (Two-Stage) on various LLM distillation datasets (RDL: our Rank-DistiLLM dataset). The highest and second-highest scores per task are bold and underlined, respectively. † denotes a statistically significant difference from the underlined monoELECTRA model ($p < 0.05$, Holm-Bonferroni-corrected).

Model	**Dataset**	**BM25**		**ColBERTv2**	
		DL 19	DL 20	DL 19	DL 20
First Stage	–	0.480†	0.494†	0.732	0.724
RankGPT-4	–	0.713	0.713	0.766	0.793
RankZephyr	RankGPT+	0.719	0.720	0.749	0.798
monoT5$_{3B}$	MS MARCO	0.705	0.715	0.745	0.757
RankT5$_{3B}$	MS MARCO	0.710	0.711	0.752	0.772
monoELECTRA$_{\text{Base}}$	MS MARCO	0.687	0.698	0.739	0.760
LLM-Distillation – Single-Stage Fine-Tuning					
monoELECTRA$_{\text{Base}}$	RankGPT	0.696	0.666†	0.690†	0.662†
monoELECTRA$_{\text{Base}}$	TWOLAR	0.693	0.669†	0.754	0.730
monoELECTRA$_{\text{Base}}$	RDL (BM25)	0.644†	0.622†	0.674†	0.654†
monoELECTRA$_{\text{Base}}$	RDL (CBv2)	0.709	0.704	**0.774**	0.754
LLM-Distillation – Two-Stage Fine-Tuning					
monoELECTRA$_{\text{Base}}$	RankGPT	0.664†	0.634†	0.477†	0.472†
monoELECTRA$_{\text{Base}}$	TWOLAR	0.715	0.706	0.763	0.760
monoELECTRA$_{\text{Base}}$	RDL (BM25)	0.672†	0.638†	0.714	0.683†
monoELECTRA$_{\text{Base}}$	RDL (CBv2)	0.720	0.711	0.768	0.770
monoELECTRA$_{\text{Large}}$	RDL (CBv2)	**0.733**	**0.727**	0.765	**0.799**

The differences to the smaller monoELECTRA$_{\text{Base}}$ model are again not significant, but the large model does improve effectiveness and is the most effective model, even outperforming RankGPT-4 and RankZephyr, in three out of four cases.

Comparison Between LLM Distillation Datasets Models fine-tuned on our ColBERTv2 Rank-DistiLLM dataset are more effective than models fine-tuned on RankGPT and TWOLAR in all cases, irrespective of single-stage or two-stage fine-tuning. Effectiveness improvements are statistically significant compared to RankGPT in all cases and compared to TWOLAR in one case.

We attribute our higher effectiveness to the fact that we use a single high-quality retrieval model for the initial retrieval when generating distillation data. Comparing the model fine-tuned on BM25 Rank-DistiLLM data to the model fine-tuned on ColBERTv2 Rank-DistiLLM data, we find that the latter is substantially more effective. The differences are statistically significant in three of four cases.

4.2 Out-of-Domain Effectiveness

Table 2 lists the effectiveness on all corpora from the TIREx framework [4–6, 9, 11–23, 26, 32, 35–37, 56, 57, 66–70]. It shows that the monoELECTRA$_{\text{Large}}$ model two-stage fine-tuned using our ColBERTv2 Rank-DistiLLM dataset is only

Table 2: Effectiveness in nDCG@10 micro-averaged across all queries from a collection from the TIREx framework [32]. The macro-averaged geometric mean is computed across all corpora. The highest and second-highest scores per corpus are bold and underlined, respectively. † denotes a statistically significant difference from the underlined monoELECTRA model ($p < 0.05$, Holm-Bonferroni-corrected). The dataset used to fine-tune each model is provided for context (RDL: our Rank-DistiLLM dataset).

Model	**Dataset**	**Antique**	**Args.me**	**ClueWeb09**	**ClueWeb12**	**CORD-19**	**Cranfield**	**Disks4+5**	**GOV**	**GOV2**	**MEDLINE**	**NFCorpus**	**Vaswani**	**WaPo**	**G. Mean**
First Stage	–	.510†	.405	.173	**.364**†	.586†	**.012**	.424†	.332†	.467†	.374	.268†	.447†	.364†	.290
RankZephyr	RGPT+	.528†	.364	.203	.303	**.767**†	.009	.542†	.423†	.560	**.453**†	.299	.512	**.508**†	.321
monoT5$_{3B}$	MSM	.537†	.392	.181	.279	.603†	.011	.545†	.407†	.514	.375	**.308**†	.458†	.476	.305
RankT5$_{3B}$	MSM	**.592**	**.421**†	**.215**	.336†	.713	.010	.538†	.415†	.528	.406	.307†	.459†	.468	**.323**
mELEC$_{\text{Base}}$	MSM	.512†	.326†	.155†	.252†	.667	.008	.436†	.337†	.491†	.364†	.255†	.456†	.406	.271
mELEC$_{\text{Base}}$	TWOLAR	.570†	.305†	.180	.292	.653†	.009	.486†	.354†	.523	.404	.267†	.519	.434	.292
mELEC$_{\text{Base}}$	RDL (CBv2)	.587	.375	.195	.295	.692	.010	.507	.388	.541	.396	.291	.522	.458	.312
mELEC$_{\text{Large}}$	RDL (CBv2)	.570†	.369	.210	.313†	.716	.008	**.546**†	**.424**†	**.572**†	.417	.301†	**.526**	.504†	.320

marginally less effective than the state-of-the-art RankZephyr and largest RankT5 models. Comparing the TWOLAR dataset to the Rank-DistiLLM dataset shows that our dataset produces a more effective model on 12 of 13 corpora and the differences are significant on 6 corpora. Our Rank-DistiLLM dataset therefore closes the effectiveness gab between distilled cross-encoders and their teacher LLMs for both in-domain and out-of-domain re-ranking.

4.3 Data Ablation

Figure 1 shows that effectiveness peaks at 50 samples per query and slightly decreases at 100 samples. We attribute the lower effectiveness at 100 samples to the training data containing fewer relevant documents at lower depths but further work is necessary to fully investigate this effect. When downsampling the number of training samples, we achieve the highest effectiveness using all 10k queries. Since monoELECTRA$_{\text{Large}}$ can reach the effectiveness of RankZephyr in two-stage fine-tuning, we assume 10k queries are sufficient in this case. However, more data may improve effectiveness in single-stage fine-tuning.

4.4 Listwise Fine-Tuning

Our newly proposed listwise ADR-MSE loss function produces a marginally less effective model at 0.002 (single-stage) and 0.001 (two-stage) lower nDCG@10 compared to using RankNet averaged across TREC DL 2019 / 2020 and BM25 / ColBERTv2 for initial retrieval. Since the effectiveness is practically equal and monoELECTRA$_{\text{Large}}$ fine-tuned using RankNet already reaches the effectiveness of RankZephyr, we conclude that listwise loss functions are (currently) unnecessary for distillation from LLMs. More complex ranking tasks may benefit from listwise losses, warranting further research for confirmation.

Fig. 1: Average effectiveness on TREC DL 2019 and 2020 for models fine-tuned on subsamples of RankDistiLLM using different depths and numbers of samples.

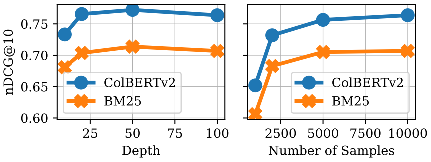

Table 3: Time in seconds and memory consumption in gigabytes for re-ranking 100 passages.

Model	**Param.**	**Time**	**Memory**
RankGPT-4	N/A	20.234	N/A
RankZephyr	7B	24.047	15.48
$\text{monoT5}_{\text{3B}}$	3B	0.998	29.36
$\text{RankT5}_{\text{3B}}$	3B	0.942	29.04
$\text{m.ELEC}_{\text{Base}}$	110M	0.139	1.18
$\text{m.ELEC}_{\text{Large}}$	330M	0.215	2.69

4.5 Efficiency

Table 3 reports the model size in parameters, latency in seconds, and GPU memory consumption in gigabytes. Our monoELECTRA models use vastly smaller backbone encoder models compared to $\text{monoT5}_{\text{3B}}$ and $\text{RankT5}_{\text{3B}}$, reducing latency and memory consumption. Our $\text{monoELECTRA}_{\text{Large}}$ model is around 4 times faster and needs around 10% of the memory compared to $\text{monoT5}_{\text{3B}}$ and $\text{RankT5}_{\text{3B}}$. While RankZephyr uses an even larger backbone encoder model, the required memory is lower than for $\text{monoT5}_{\text{3B}}$ and $\text{RankT5}_{\text{3B}}$ because it only re-ranks 20 passages at a time. The model still requires around 5.7 times the amount of memory as $\text{monoELECTRA}_{\text{Large}}$. Furthermore, the windowed ranking strategy necessitates that some passages are scored multiple times, leading to very poor latency. RankZephyr is around 110 times slower than $\text{monoELECTRA}_{\text{Large}}$ at comparable effectiveness. This poor latency also applies to RankGPT-4 since it uses the same ranking strategy.

5 Conclusion

Using our new Rank-DistiLLM datset, we have systematically investigated several aspects of distilling cross-encoders from LLM rankings. Our findings indicate that rankings of the top-50 passages for 10,000 queries suffice to achieve competitive effectiveness compared to LLMs, but the passages need to be sampled using a very effective first-stage retrieval model. By first fine-tuning on MS MARCO labels and then further on Rank-DistiLLM, our best model is more effective than previous cross-encoders and matches the effectiveness of LLMs for in- and out-of-domain re-ranking while being orders of magnitude more efficient.

Acknowledgements This publication has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101070014 (OpenWebSearch.EU, https://doi.org/10.3030/101070014).

References

[1] Akkalyoncu Yilmaz, Z., Yang, W., Zhang, H., Lin, J.: Cross-Domain Modeling of Sentence-Level Evidence for Document Retrieval. In: Proceedings of EMNLP-IJCNLP 2019, pp. 3490–3496 (2019), https://doi.org/10.18653/v1/D19-1352

[2] Arabzadeh, N., Vtyurina, A., Yan, X., Clarke, C.L.A.: Shallow Pooling for Sparse Labels. Information Retrieval Journal **25**, 365–385 (2022), ISSN 1573-7659, https://doi.org/10.1007/s10791-022-09411-0

[3] Baldelli, D., Jiang, J., Aizawa, A., Torroni, P.: TWOLAR: A TWO-Step LLM-Augmented Distillation Method for Passage Reranking. In: Proceedings of ECIR 2024, pp. 470–485 (2024), https://doi.org/10.1007/978-3-031-56027-9_29

[4] Bondarenko, A., Fröbe, M., Kiesel, J., Syed, S., Gurcke, T., Beloucif, M., Panchenko, A., Biemann, C., Stein, B., Wachsmuth, H., Potthast, M., Hagen, M.: Overview of Touché 2022: Argument Retrieval. In: Proceedings of CLEF 2022, pp. 311–336 (2022), https://doi.org/10.1007/978-3-031-13643-6_21

[5] Bondarenko, A., Gienapp, L., Fröbe, M., Beloucif, M., Ajjour, Y., Panchenko, A., Biemann, C., Stein, B., Wachsmuth, H., Potthast, M., Hagen, M.: Overview of Touché 2021: Argument Retrieval. In: Proceedings of CLEF 2021, pp. 450–467 (2021), https://doi.org/10.1007/978-3-030-85251-1_28

[6] Boteva, V., Gholipour, D., Sokolov, A., Riezler, S.: A Full-Text Learning to Rank Dataset for Medical Information Retrieval. In: Proceedings of ECIR 2016, pp. 716–722 (2016), https://doi.org/10.1007/978-3-319-30671-1_58

[7] Bruch, S., Wang, X., Bendersky, M., Najork, M.: An Analysis of the Softmax Cross Entropy Loss for Learning-to-Rank with Binary Relevance. In: Proceedings of SIGIR 2019, pp. 75–78 (2019), https://doi.org/10.1145/3341981.3344221

[8] Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to Rank Using Gradient Descent. In: Proceedings of ICML 2005, pp. 89–96 (2005), https://doi.org/10.1145/1102351.1102363

[9] Büttcher, S., Clarke, C.L.A., Soboroff, I.: The TREC 2006 Terabyte Track. In: Proceedings of TREC 2006 (2006), URL http://trec.nist.gov/pubs/trec15/papers/TERA06.OVERVIEW.pdf

[10] Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In: Proceedings of ICLR 2020 (2020), URL https://openreview.net/forum?id=r1xMH1BtvB

[11] Clarke, C.L.A., Craswell, N., Soboroff, I.: Overview of the TREC 2004 Terabyte Track. In: Proceedings of TREC 2004 (2004), URL http://trec.nist.gov/pubs/trec13/papers/TERA.OVERVIEW.pdf

[12] Clarke, C.L.A., Craswell, N., Soboroff, I.: Overview of the TREC 2009 Web Track. In: Proceedings of TREC 2009 (2009), URL http://trec.nist.gov/pubs/trec18/papers/WEB09.OVERVIEW.pdf

[13] Clarke, C.L.A., Craswell, N., Soboroff, I., Cormack, G.V.: Overview of the TREC 2010 Web Track. In: Proceedings of TREC 2010 (2010), URL https://trec.nist.gov/pubs/trec19/papers/WEB.OVERVIEW.pdf

[14] Clarke, C.L.A., Craswell, N., Soboroff, I., Voorhees, E.M.: Overview of the TREC 2011 Web Track. In: Proceedings of TREC 2011 (2011), URL http://trec.nist.gov/pubs/trec20/papers/WEB.OVERVIEW.pdf

[15] Clarke, C.L.A., Craswell, N., Voorhees, E.M.: Overview of the TREC 2012 Web Track. In: Proceedings of TREC 2012 (2012), URL http://trec.nist.gov/pubs/trec21/papers/WEB12.overview.pdf

[16] Clarke, C.L.A., Scholer, F., Soboroff, I.: The TREC 2005 Terabyte Track. In: Proceedings of TREC 2005 (2005), URL http://trec.nist.gov/pubs/trec14/papers/TERABYTE.OVERVIEW.pdf

[17] Cleverdon, C.W.: The Significance of the Cranfield Tests on Index Languages. In: Proceedings of SIGIR 1991, pp. 3–12 (1991), https://doi.org/10.1145/122860.122861

[18] Collins-Thompson, K., Bennett, P.N., Diaz, F., Clarke, C., Voorhees, E.M.: TREC 2013 Web Track Overview. In: Proceedings of TREC 2013 (2013), URL http://trec.nist.gov/pubs/trec22/papers/WEB.OVERVIEW.pdf

[19] Collins-Thompson, K., Macdonald, C., Bennett, P.N., Diaz, F., Voorhees, E.M.: TREC 2014 Web Track Overview. In: Proceedings of TREC 2014 (2014), URL http://trec.nist.gov/pubs/trec23/papers/overview-web.pdf

[20] Craswell, N., Hawking, D.: Overview of the TREC-2002 Web Track. In: Proceedings of TREC 2002 (2002), URL http://trec.nist.gov/pubs/trec11/papers/WEB.OVER.pdf

[21] Craswell, N., Hawking, D.: Overview of the TREC 2004 Web Track. In: Proceedings of TREC 2004 (2004), URL http://trec.nist.gov/pubs/trec13/papers/WEB.OVERVIEW.pdf

[22] Craswell, N., Hawking, D., Wilkinson, R., Wu, M.: Overview of the TREC 2003 Web Track. In: Proceedings of TREC 2003, pp. 78–92 (2003), URL http://trec.nist.gov/pubs/trec12/papers/WEB.OVERVIEW.pdf

[23] Craswell, N., Mitra, B., Yilmaz, E., Campos, D.: Overview of the TREC 2020 Deep Learning Track. In: Proceedings of TREC 2020 (2020), URL https://trec.nist.gov/pubs/trec29/papers/OVERVIEW.DL.pdf

[24] Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Lin, J.: Overview of the TREC 2021 Deep Learning Track. In: Proceedings of TREC 2021 (2021), URL https://trec.nist.gov/pubs/trec30/papers/Overview-DL.pdf

[25] Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Lin, J., Voorhees, E.M., Soboroff, I.: Overview of the TREC 2022 Deep Learning Track. In: Proceedings of TREC 2022 (2022), URL https://trec.nist.gov/pubs/trec31/papers/Overview_deep.pdf

[26] Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Voorhees, E.M.: Overview of the TREC 2019 Deep Learning Track. In: Proceedings of TREC 2019 (2019), URL https://trec.nist.gov/pubs/trec28/papers/OVERVIEW.DL.pdf

[27] Dao, T.: FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. arXiv:2307.08691 (2023), https://doi.org/10.48550/arXiv.2307.08691

[28] Dao, T., Fu, D.Y., Ermon, S., Rudra, A., Ré, C.: FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. In: Proceedings of NeurIPS 2022, pp. 16344–16359 (2022), URL http://papers.nips.cc/paper_files/paper/2022/hash/67d57c32e20fd0a7a302cb81d36e40d5-Abstract-Conference.html

[29] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of NAACL 2019, pp. 4171–4186 (2019), https://doi.org/10.18653/v1/N19-1423

[30] Falcon, W., The PyTorch Lightning team: PyTorch Lightning (2023), https://doi.org/10.5281/zenodo.7859091

[31] Formal, T., Piwowarski, B., Clinchant, S.: SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking. In: Proceedings of SIGIR 2021, pp. 2288–2292 (2021), https://doi.org/10.1145/3404835.3463098

[32] Fröbe, M., Reimer, J.H., MacAvaney, S., Deckers, N., Reich, S., Bevendorff, J., Stein, B., Hagen, M., Potthast, M.: The Information Retrieval Experiment Platform. In: Proceedings of SIGIR 2023, pp. 2826–2836 (2023), https://doi.org/10.1145/3539618.3591888

[33] Gao, L., Dai, Z., Callan, J.: Rethink Training of BERT Rerankers in Multi-stage Retrieval Pipeline. In: Proceedings of ECIR 2021, pp. 280–286 (2021), https://doi.org/10.1007/978-3-030-72240-1_26

[34] Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E.: Array Programming with NumPy. Nature **585**(7825), 357–362 (2020), https://doi.org/10.1038/s41586-020-2649-2

[35] Hashemi, H., Aliannejadi, M., Zamani, H., Croft, W.B.: ANTIQUE: A Non-factoid Question Answering Benchmark. In: Proceedings of ECIR 2020, pp. 166–173 (2020), https://doi.org/10.1007/978-3-030-45442-5_21

[36] Hersh, W.R., Bhupatiraju, R.T., Ross, L., Cohen, A.M., Kraemer, D., Johnson, P.: TREC 2004 Genomics Track Overview. In: Proceedings of TREC 2004 (2004), URL http://trec.nist.gov/pubs/trec13/papers/GEO.OVERVIEW.pdf

[37] Hersh, W.R., Cohen, A.M., Yang, J., Bhupatiraju, R.T., Roberts, P.M., Hearst, M.A.: TREC 2005 Genomics Track Overview. In: Proceedings of TREC 2005 (2005), URL http://trec.nist.gov/pubs/trec14/papers/GEO.OVERVIEW.pdf

[38] Hofstätter, S., Althammer, S., Schröder, M., Sertkan, M., Hanbury, A.: Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation. arXiv:2010.02666 (2021), https://doi.org/10.48550/arXiv.2010.02666

[39] Hunter, J.D.: Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering **9**(03), 90–95 (May 2007), ISSN 1521-9615, https://doi.org/10.1109/MCSE.2007.55

[40] Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C., Team, J.D.: Jupyter Notebooks – A Publishing Format for Reproducible Computational Workflows. In: Positioning and Power in Academic Publishing: Players, Agents and Agendas, pp. 87–90, IOS Press (2016), https://doi.org/10.3233/978-1-61499-649-1-87

[41] Lefaudeux, B., Massa, F., Liskovich, D., Xiong, W., Caggiano, V., Naren, S., Xu, M., Hu, J., Tintore, M., Zhang, S., Labatut, P., Haziza, D.: xFormers: A modular and hackable Transformer modelling library. https://github.com/facebookresearch/xformers (2022)

[42] Lin, S.C., Asai, A., Li, M., Oguz, B., Lin, J., Mehdad, Y., Yih, W.t., Chen, X.: How to Train Your Dragon: Diverse Augmentation Towards Generalizable Dense Retrieval. In: Findings of EMNLP 2023, pp. 6385–6400 (2023), https://doi.org/10.18653/V1/2023.FINDINGS-EMNLP.423

[43] Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization. In: Proceedings of ICLR 2019 (2019), URL https://openreview.net/forum?id=Bkg6RiCqY7

[44] MacAvaney, S., Macdonald, C., Ounis, I.: Streamlining Evaluation with `ir-measures`. In: Proceedings of ECIR 2022, pp. 305–310 (2022), https://doi.org/10.1007/978-3-030-99739-7_38

[45] MacAvaney, S., Yates, A., Feldman, S., Downey, D., Cohan, A., Goharian, N.: Simplified Data Wrangling with `ir_datasets`. In: Proceedings of SIGIR 2021, pp. 2429–2436 (2021), https://doi.org/10.1145/3404835.3463254

[46] Macdonald, C., Tonellotto, N., MacAvaney, S., Ounis, I.: PyTerrier: Declarative Experimentation in Python from BM25 to Dense Retrieval. In: Proceedings of CIKM 2021, pp. 4526–4533 (2021), https://doi.org/10.1145/3459637.3482013

[47] Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In: Proceedings of COCO@NeurIPS 2016 (2016), URL https://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf

[48] Nogueira, R., Cho, K.: Passage Re-ranking with BERT. arXiv:1901.04085[v5] (2020), https://doi.org/10.48550/arXiv.1901.04085

[49] Nogueira, R., Jiang, Z., Pradeep, R., Lin, J.: Document Ranking with a Pretrained Sequence-to-Sequence Model. In: Findings of EMNLP 2020, pp. 708–718 (2020), https://doi.org/10.18653/v1/2020.findings-emnlp.63

[50] van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv:1807.03748 (2019), https://doi.org/10.48550/arXiv.1807.03748

[51] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Proceedings of NeurIPS 2019, pp. 8024–8035 (2019), URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html

[52] Pradeep, R., Liu, Y., Zhang, X., Li, Y., Yates, A., Lin, J.: Squeezing Water from a Stone: A Bag of Tricks for Further Improving Cross-Encoder Effectiveness for Reranking. In: Proceedings of ECIR 2022, pp. 655–670 (2022), https://doi.org/10.1007/978-3-030-99736-6_44

[53] Pradeep, R., Sharifymoghaddam, S., Lin, J.: RankVicuna: Zero-Shot Listwise Document Reranking with Open-Source Large Language Models. arXiv:2309.15088 (2023), URL https://doi.org/10.48550/arXiv.2309.15088

[54] Pradeep, R., Sharifymoghaddam, S., Lin, J.: RankZephyr: Effective and Robust Zero-Shot Listwise Reranking is a Breeze! arXiv:2312.02724 (2023), URL https://doi.org/10.48550/arXiv.2312.02724

[55] Qin, T., Liu, T.Y., Li, H.: A general approximation framework for direct optimization of information retrieval measures. Information Retrieval **13**, 375–397 (2010), https://doi.org/10.1007/s10791-009-9124-x

[56] Roberts, K., Demner-Fushman, D., Voorhees, E.M., Hersh, W.R., Bedrick, S., Lazar, A.J.: Overview of the TREC 2018 Precision Medicine Track. In: Proceedings of TREC 2018 (2018), URL https://trec.nist.gov/pubs/trec27/papers/Overview-PM.pdf

[57] Roberts, K., Demner-Fushman, D., Voorhees, E.M., Hersh, W.R., Bedrick, S., Lazar, A.J., Pant, S.: Overview of the TREC 2017 Precision Medicine Track. In: Proceedings of TREC 2017 (2017), URL https://trec.nist.gov/pubs/trec26/papers/Overview-PM.pdf

[58] Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at TREC-3. In: Proceedings of TREC 1994, pp. 109–126 (1994), URL http://trec.nist.gov/pubs/trec3/papers/city.ps.gz

[59] Rosa, G., Bonifacio, L., Jeronymo, V., Abonizio, H., Fadaee, M., Lotufo, R., Nogueira, R.: In Defense of Cross-Encoders for Zero-Shot Retrieval. arXiv:2212.06121 (2022), https://doi.org/10.48550/arXiv.2212.06121

[60] Santhanam, K., Khattab, O., Saad-Falcon, J., Potts, C., Zaharia, M.: ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. arXiv:2112.01488 (2022), https://doi.org/10.48550/arXiv.2112.01488

[61] Schlatt, F., Fröbe, M., Hagen, M.: Lightning IR: Straightforward Fine-tuning and Inference of Transformer-based Language Models for Information Retrieval. In:

Proceedings of WSDM 2025 (2025), https://doi.org/10.1145/3701551.3704118, (to appear)

[62] Sun, W., Yan, L., Ma, X., Wang, S., Ren, P., Chen, Z., Yin, D., Ren, Z.: Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents. In: Proceedings of EMNLP 2023, pp. 14918–14937 (2023), https://doi.org/10.18653/v1/2023.emnlp-main.923

[63] Tamber, M.S., Pradeep, R., Lin, J.: Scaling Down, LiTting Up: Efficient Zero-Shot Listwise Reranking with Seq2seq Encoder-Decoder Models. arXiv:2312.16098 (2023), https://doi.org/10.48550/arXiv.2312.16098

[64] pandas development team, T.: Pandas-dev/pandas: Pandas. Zenodo (Apr 2024), https://doi.org/10.5281/zenodo.10957263

[65] Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P.: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods **17**(3), 261–272 (2020), https://doi.org/10.1038/s41592-019-0686-2

[66] Voorhees, E., Alam, T., Bedrick, S., Demner-Fushman, D., Hersh, W.R., Lo, K., Roberts, K., Soboroff, I., Wang, L.L.: TREC-COVID: Constructing a Pandemic Information Retrieval Test Collection. arXiv:2005.04474 (2020), https://doi.org/10.48550/arXiv.2005.04474

[67] Voorhees, E.M.: Overview of the TREC 2004 Robust Track. In: Procedings of TREC 2004 (2004), URL http://trec.nist.gov/pubs/trec13/papers/ROBUST.OVERVIEW.pdf

[68] Voorhees, E.M., Harman, D.: Overview of the Seventh Text REtrieval Conference (TREC-7). In: Proceedings of TREC 1998 (1998), URL http://trec.nist.gov/pubs/trec8/papers/overview_8.ps

[69] Voorhees, E.M., Harman, D.: Overview of the Eigth Text REtrieval Conference (TREC-8). In: Proceedings of TREC 1999 (1999), URL http://trec.nist.gov/pubs/trec8/papers/overview_8.ps

[70] Wang, L.L., Lo, K., Chandrasekhar, Y., Reas, R., Yang, J., Burdick, D., Eide, D., Funk, K., Katsis, Y., Kinney, R., Li, Y., Liu, Z., Merrill, W., Mooney, P., Murdick, D., Rishi, D., Sheehan, J., Shen, Z., Stilson, B., Wade, A., Wang, K., Wang, N.X.R., Wilhelm, C., Xie, B., Raymond, D., Weld, D.S., Etzioni, O., Kohlmeier, S.: CORD-19: The COVID-19 Open Research Dataset. arXiv:2004.10706 (2020), https://doi.org/10.48550/arXiv.2004.10706

[71] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:1910.03771[v5] (2020), https://doi.org/10.48550/arXiv.1910.03771

[72] Xiong, L., Xiong, C., Li, Y., Tang, K.F., Liu, J., Bennett, P.N., Ahmed, J., Overwijk, A.: Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In: Proceedings of ICLR 2021 (2021), URL https://openreview.net/forum?id=zeFrfgyZln

[73] Zhuang, H., Qin, Z., Jagerman, R., Hui, K., Ma, J., Lu, J., Ni, J., Wang, X., Bendersky, M.: RankT5: Fine-Tuning T5 for Text Ranking with Ranking Losses. arXiv:2210.10634 (2022), https://doi.org/10.48550/arXiv.2210.10634