CELI: A Simple yet Effective Approach to Enhance Out-of-Domain Generalization of Cross-Encoders

Xinyu Zhang,* **Minghan Li,*** **Jimmy Lin**
David R. Cheriton School of Computer Science, University of Waterloo, Canada
{x978zhan, m692li, jimmylin}@uwaterloo.ca

Abstract

In text ranking, it is generally believed that the cross-encoders already gather sufficient token interaction information via the attention mechanism in the hidden layers. However, our results show that the cross-encoders can consistently benefit from additional token interaction in the similarity computation at the last layer. We introduce **CELI** (**C**ross-**E**ncoder with **L**ate **I**nteraction), which incorporates a late interaction layer into the current cross-encoder models. This simple method brings 5% improvement on BEIR without compromising in-domain effectiveness or search latency. Extensive experiments show that this finding is consistent across different sizes of the cross-encoder models and the first-stage retrievers. Our findings suggest that boiling all information into the [CLS] token is a suboptimal use for cross-encoders, and advocate further studies to investigate its relevance score mechanism.

Model	**MS MARCO** MRR@10	**BEIR Avg.** nDCG@10	**Search Latency**
monoBERT	0.390	0.467	1.18s
CELI	0.392	0.491	1.28s

Table 1: A preview of comparison between CELI and monoBERT. Detailed results are shown in Table 2 and 3.

1 Introduction

The two-stage retrieve-then-rerank pipeline has been the *de facto* design for many information retrieval systems. Recently, both the retriever and the reranker systems have benefited from the advancement in pretrained language models (Lin et al., 2022). When using the pretrained models as cross-encoders, the model views the query and document candidate together, allowing rich token interaction via the attention mechanism at all hidden layers. However, all the information then boils down to the representation of [CLS] token at the final stage, during the computation of the relevance score between the query and the document. This raises the concern of whether the single token representation is sufficient to capture all salient information.

Pretrained language models have also been adopted in the retriever stage in various ways. Karpukhin et al. (2020) pioneer in this direction and find that the [CLS] token embedding could well capture query or document representations, whose similarity can be used to indicate the relevance level between the query and document. This line of methods (Karpukhin et al., 2020; Xiong et al., 2021), named as *single-vector dense retrievers*, while being effective for the in-domain scenarios, is found to be less robust on the out-of-distribution (OOD) datasets (Thakur et al., 2021), possibly due to inadequate information at token-level granularity. Methods such as further pretraining or adding token-level interaction have been applied to improve the OOD generalization, where *multi-vector retrievers* (Khattab and Zaharia, 2020; Santhanam et al., 2022; Li et al., 2023) perform the best on both the in-domain and OOD effectiveness among neural retrievers (Lin et al., 2023a). This ability is usually credited to its design that computes the similarity score based on contextualized embeddings of all tokens, which provides richer token interactions compared to the single-vector dense retrievers.

Inspired by the success of token interaction in the retriever systems, we ask the question: *Can cross-encoder also benefit from additional token interaction when computing the final similarity?* In this work, we affirm this hypothesis, showing that additional token interaction in the final-stage similarity computation indeed improves the OOD capacity for cross-encoders. We name our method **CELI** (**C**ross-**E**ncoder with **L**ate **I**nteraction), which incorporates a late interaction layer into the current

* Equal Contribution

cross-encoder models. As shown in Table 1, CELI improves averaged nDCG@10 on BEIR by 5% (from 0.467 to 0.491), while not sacrificing the in-domain score (0.390 vs. 0.392) and the search latency (1.18s vs. 1.28s). Extensive experiments show that the improvement is consistent over larger-sized models and reranking candidates from various retrievers.

2 Methods

monoBERT. monoBERT (Nogueira and Cho, 2019) is one of the first cross-encoders (MacAvaney et al., 2019; Nogueira et al., 2020) that apply pretrained transformers in passage retrieval. Given concatenated query q and document d, the model computes relevance scores $s_{q,d}$ from the `[CLS]` representation on the final layer of BERT (Devlin et al., 2019), formulated as follows (Lin et al., 2022; Pradeep et al., 2022):

$$s_m(q,d) = T_{\texttt{[CLS]}} W + b, \tag{1}$$

where $T_{\texttt{[CLS]}} \in \mathbf{R}^D$ is the `[CLS]` representation on the final layer, and $W \in \mathbf{R}^{D\times 1}$ and $b \in \mathbf{R}$ are the weight and bias for classification.

Some of the previous works term the models as "mono{`BACKBONE`}" when initialized from non-BERT pretrained Transformers, such as monoELECTRA (Pradeep et al., 2022). However, since the underlying model structure remains the same, we refer to them all as monoBERT while specifying the backbones where the models are initialized.

Mean-Pooling. To study whether the improvement of CELI is attributed to the interaction between the query and the documents tokens, or simply the additional token information, we add the Mean-Pooling method as a baseline. Instead of computing the similarity score based solely on the `[CLS]` representation as in Eq. (1), it uses the mean representation of all the tokens:

$$s_m(q,d) = \frac{1}{n}\sum_{i}^{n}\left(T_{tok_i} W + b\right), \tag{2}$$

where T_{tok_i} is the final-layer representation of the i-th token, and n is the total number of tokens in the input sequence. $W \in \mathbf{R}^{D\times 1}$ and $b \in \mathbf{R}$ are the weight and bias for classification, same as Eq. (1).

CELI. In this work, we use the simplest version of late interaction proposed by Khattab and Zaharia (2020). We first obtain the representation of each token in the query q and document d:

$$v_{q_i} = T_{q_i} W + b; \quad v_{d_j} = T_{d_j} W + b, \tag{3}$$

where q_i and d_j represent the i-th token of query q and the j-th token of document d, respectively. Similar to Eq. (1), $T \in \mathbf{R}^D$ refers to each token representation on the final layer. $W \in \mathbf{R}^{D\times D_{tok}}$ and $b \in \mathbf{R}^{D_{tok}}$ are the weight and bias in a projection layer, projecting the T_{tok} to a lower dimension $D_{tok} < D$.

With token representations v_{q_i} and v_{d_j}, the late interaction first computes the inner product scores between all pairs of query and document tokens, then sums up the max similarity score for each query token against all document tokens:

$$s_l(q,d) = \sum_{q_i} \max_{d_j}(v_{q_i}^T v_{d_j}). \tag{4}$$

Eq. (4) shares the same formulation as in the first-stage retrievers, and only differ in that the token representations T_{q_i} and T_{d_j} embed information from both the query and document, whereas in first-stage retrievers, they are computed independently from each other, with T_{q_i} perceiving no information from document d and vice versa.

During training, we compute the LCE loss on s_m and s_l, respectively:

$$\mathcal{L} = lce(s_m(q,d^+), s_m(q,d_1^-), ..., s_m(q,d_n^-)) \\ + lce(s_l(q,d^+), s_l(q,d_1^-), ..., s_l(q,d_n^-)),$$

where d^+ is the positive document and $\{d_i^-\}_{i=1}^n$ are the negative documents to the query q.

At inference time, we sum the two scores as the final relevance score, i.e., $s_{final} = s_m + s_l$.[1]

3 Experimental Setup

All cross-encoders are trained on MS MARCO (Bajaj et al., 2016), a dataset composed of queries from Bing search log and a collection of passages sourced from the general Web. It contains 8.8M passages, over 500k query–document pairs for training, and 6980 queries for inference.

We implement the model based on Capreolus (Yates et al., 2020a,b), an IR toolkit for end-to-end neural ad hoc retrieval. All training configurations follow Pradeep et al. (2022): We train MS

[1]We have explored adding weighting terms for s_m and s_c, but only observed marginal gains. Thus we report the simplest formulation here.

Backbone	Model	MS MARCO MRR@10	BEIR nDCG@10
MiniLM	monoBERT	0.390	0.467
	Mean-Pooling	0.390	0.481
	CELI	0.392	**0.491**
ELECTRA$_{base}$	monoBERT	0.400	0.481
	Mean-Pooling	0.402	0.483
	CELI	0.402	**0.494**
ELECTRA$_{large}$	monoBERT	0.413	0.507
	Mean-Pooling	0.412	0.516
	CELI	0.413	**0.524**

Table 2: In-domain (MRR@10 on MS MARCO) and OOD (averaged nDCG@10 on BEIR) scores of CELI and two baselines (i.e., monoBERT and Mean-Pooling). *Detailed scores on BEIR are reported in Table 5.

Model	*Sparse*			*Multi-vector Dense*
	BM25	**uniCOIL**	**SPLADE**	**ColBERT v2**
monoBERT	0.467	0.426	0.469	0.467
CELI	0.491	0.452	0.492	0.493

Model	*Single-vector Dense*				
	DPR (NQ)	**DPR (MS)**	**ANCE**	**TCT**	**TAS-B**
monoBERT	0.451	0.474	0.471	0.470	0.472
CELI	0.472	0.495	0.493	0.494	0.494

Table 3: Averaged nDCG@10 on BEIR, reranking the top-1k candidates from each retriever. **TCT**: TCT-ColBERT. **DPR (NQ/MS)**: DPR fine-tuned on NQ (Kwiatkowski et al., 2019) or MS MARCO, respectively. *Detailed scores on BEIR are reported in Table 6.

MARCO for 30k steps with a learning rate $1e-5$ and a batch size 16. We use linear warmup on the first 3k steps, then linearly decay the learning rate on the rest of the steps. Cross-encoders are trained on LCE loss (Gao et al., 2021b; Pradeep et al., 2022) with 7 negative samples.[2] We experimented with three backbones, all available on HuggingFace (Wolf et al., 2020): MiniLM (Wang et al., 2020),[3] ELECTRA$_{base}$,[4] and ELECTRA$_{large}$ (Clark et al., 2020).[5]

We use MS MARCO (Bajaj et al., 2016) for the in-domain evaluation and 13 datasets from BEIR (Thakur et al., 2021) for OOD evaluation, which covers 10 domains including Wikipedia, Finance, Scientific, Quora, and so on.

At the inference stage, we always rerank top-$1k$ results from the first-stage retrievers. On MS MARCO, we use TCT-ColBERT (Lin et al., 2021b) as the retriever following Pradeep et al. (2022). On BEIR, we use an extensive list of retrievers that covers the categories of sparse, single- and multi-vector dense retrievers. Retrievers results are produced using Pyserini (Lin et al., 2021a), BEIR (Thakur et al., 2021), or ColBERT (Khattab and Zaharia, 2020) repository.[6] Following the datasets standard, we report MRR@10 on MS MARCO and nDCG@10 on BEIR.

[2]We use Quadro RTX 8000 GPUs and A6000 for the experiments. On RTX 8000, the ELECTRA$_{base}$ models took approximately 8 hours for cross-encoder training.

[3]microsoft/MiniLM-L12-H384-uncased

[4]google/electra-base-discriminator

[5]google/electra-large-discriminator

[6]https://github.com/stanford-futuredata/ColBERT

4 Results and Analysis

Table 1 provides a preview of the efficacy of CELI. In this section, we first demonstrate that such improvement is consistent over different model sizes and the first-stage retrievers, then analyze how the projected token dimension and the query length impact the improvement.

4.1 Model Size

Previous papers find that models with a larger number of parameters can better generate on unseen distribution (Ni et al., 2022). Motivated by this observation, we examine whether the improvement brought by late interaction diminishes with increasing model sizes.

Results show that the contribution of late interaction is consistent over model size. Table 2 shows in-domain and OOD scores with the models initialized from three different sizes of backbones: MiniLM, ELECTRA$_{base}$, and ELECTRA$_{large}$.[7] The MS MARCO results reranks the top-1k candidates from TCT-ColBERT, and the BEIR results reranks the top-1k candidates from BM25.

While we observe higher average scores on BEIR as the model size increases, echoing the previous finding that larger models demonstrate better OOD generalization ability, the improvement brought by token information is consistent across the backbones. On all three models, CELI consistently improves over the two baselines. Additionally, the in-domain scores on the other two backbones are not affected as well, suggesting that the "free" gain is consistent over different model sizes.

[7]MiniLM, ELECTRA$_{base}$, and ELECTRA$_{large}$ have 33M, 110M, and 340M parameters respectively.

	Projected Token Dimension (D_{tok})	**MS MARCO** MRR@10	**BEIR** nDCG@10
(1)	$D_{\text{tok}} = 1$	0.3920	0.4890
(2)	$D_{\text{tok}} = 32$	0.3920	0.4914
(3)	$D_{\text{tok}} = 128$	0.3920	0.4910
(4)	$D_{\text{tok}} = 384$	0.3900	0.4911

Table 4: MRR@10 on MS MARCO and nDCG@10 on BEIR of CELI with token representation in different dimensions (D_{tok} in Eq. (3)). We report scores to 4 digits here as the values are close in all conditions.

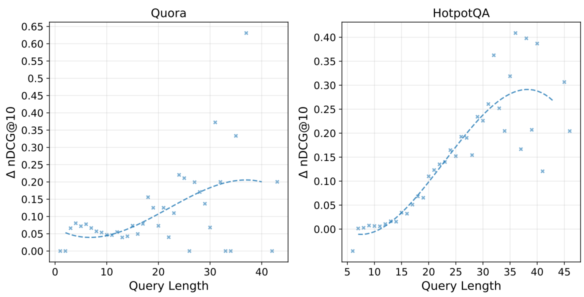

Figure 1: nDCG@10 improvement from late interaction on queries over different lengths. Each point represents the average of nDCG@10 improvements over the query of the corresponding length. The line is the least square polynomial fit of the points.

4.2 First-Stage Retriever

We then extend the experiments into an extensive list of first-stage retrievers, where the retrievers are categorized as sparse, single-vector, and multi-vector dense retrievers.

Table 3 shows the results on BEIR reranking candidates from 9 different retrievers, covering all three categories mentioned above. Looking at the averaged nDCG@10 on BEIR, we find that late interaction consistently improves the OOD capacity when using retrievers of different natures, bringing a similar degree of improvement of 0.02–0.03.

4.3 Token Dimensions

In first-stage retrieval, it is common to project the token representation into lower dimensions as restricted by indexing storage space and search efficiency. However, the representations are computed on the fly for cross-encoders, thus using token representations in higher dimensions brings no additional storage cost and only minor searching latency in the context of cross-encoders. We therefore examine whether using higher token dimensions D_{tok} brings additional improvements.

Results are shown in Table 4, where row (2) corresponds to the BM25 results reported in Table 3. Comparing rows (1–4), we find that the token dimensions have little impact on the OOD effectiveness: surprisingly, using $dim = 1$ already obtains 0.4890 on average BEIR as shown in on row (1), while increasing the dimension to $dim = 32$ and onwards only provides marginal improvement.

4.4 Query Length

Finally, we present our analysis of how the late interaction improves the OOD capacity of cross-encoders, finding that query length is a prominent indicator of the per-query improvement. Figure 1 plots the distribution of nDCG@10 improvement by late interaction according to the query length on Quora and HotpotQA, two datasets included in BEIR.[8] Specifically, each point represents the average of nDCG@10 improvements over the query of the same length (same coordinate on the x-axis). We additionally plot an approximated polynomial line based on the points to better reveal the relationship between the query length and the improvement on nDCG@10.

On both datasets, we observe a clear tendency that the late interaction brings higher improvement on longer queries. While Figure 1 is based on results using BM25 as the retriever, we have similar observations when reranking candidates from the other retrievers.

5 Related Work

Nogueira and Cho (2019) is one of the first cross-encoders that apply pretrained language models on the passage retrieval task. It considers retrieval as a classification task and uses transformer encoders following the formulation of the next sentence prediction (NSP) pretraining task in BERT, where only the `[CLS]` vector is used to classify the query–document pair and compute the relevant score. Afterward, CEDR (MacAvaney et al., 2019) proposes to incorporate fine-grained token interaction. However, it requires extra complex computations at all layers, which brings difficulty to implementation and adds higher computational overhead.

This line of cross-encoders has been studied and extended to other model architectures: Nogueira et al. (2020) and Zhuang et al. (2023) build cross-encoders on encoder-decoder architecture (e.g., T5, Raffel et al., 2020), and Ma et al. (2023) extend it

[8] Length determined as the number of query tokens delimited by whitespace.

to decoder-only architecture (e.g. LLaMA-2, Touvron et al., 2023). Another line of cross-encoders reranks the document candidates according to the query likelihood given a passage, usually based on generative models (Nogueira dos Santos et al., 2020; Sachan et al., 2022).

Recent works on first-stage retrieval have demonstrated the effectiveness of adding sparse information into dense retrieval (Chen et al., 2022). The combination of the token information and dense `[CLS]` vector could also be done explicitly, by either adding the scores computed from `[CLS]` and token information or concatenating aggregated token vectors to the `[CLS]` vector (Gao et al., 2021a; Lin et al., 2023b). The multi-vector dense models could also be viewed under this category, where the token representation vectors jointly contribute to the relevancy computation along with the `[CLS]` vector (Khattab and Zaharia, 2020; Li et al., 2023).

Our work is also connected to the interaction-based methods predating pretrained language models, where the text relevance is usually predicted based on the fine-grained similarity matrix between queries and document tokens (Socher et al., 2011; Lu and Li, 2013; Hu et al., 2014; Pang et al., 2016).

6 Conclusion

In this work, we show that adding late interaction to existing cross-encoders brings visible improvement to its OOD capacity without hurting in-domain effectiveness, even though the cross-encoder already processes the token interaction in earlier layers. Extensive experiments on different model sizes and first-stage retrievers show that this improvement is consistent, and according to our analysis, the improvement is more prominent on longer queries. Our findings suggest that boiling all information into the `[CLS]` token is a suboptimal use for cross-encoders, and further studies are required to better explore their capacities.

7 Limitations

While CELI serves as a simple yet effective approach to improve the OOD generalization capacity for cross-encoders, it is not a novel architectural innovation. Instead, it draws inspiration from first-stage retrievers (Khattab and Zaharia, 2020). That said, We prioritize this simple design approach because we value ease of use and simplicity over novelty in this context.

8 Acknowledgement

This research was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *ArXiv*, abs/1611.09268.

Xilun Chen, Kushal Lakhotia, Barlas Oguz, Anchit Gupta, Patrick Lewis, Stan Peshterliev, Yashar Mehdad, Sonal Gupta, and Wen-tau Yih. 2022. Salient phrase aware dense retrieval: Can a dense retriever imitate a sparse one? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 250–262, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021a. 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*, Online. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021b. Rethink training of BERT rerankers in multi-stage retrieval pipeline. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 – April 1, 2021, Proceedings, Part II*, page 280–286, Berlin, Heidelberg. Springer-Verlag.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. *ArXiv*, abs/1503.03244.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and

Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020)*, pages 39–48.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Minghan Li, Sheng-Chieh Lin, Barlas Oguz, Asish Ghoshal, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. CITADEL: Conditional token interaction via dynamic lexical routing for efficient and effective multi-vector retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11891–11907, Toronto, Canada. Association for Computational Linguistics.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021a. 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)*, pages 2356–2362.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2022. *Pretrained transformers for text ranking: BERT and beyond*. Springer Nature.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023a. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6385–6400, Singapore. Association for Computational Linguistics.

Sheng-Chieh Lin, Minghan Li, and Jimmy Lin. 2023b. Aggretriever: A simple approach to aggregate textual representations for robust dense passage retrieval. *Transactions of the Association for Computational Linguistics*, 11:436–452.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021b. 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)*, pages 163–173, Online. Association for Computational Linguistics.

Zhengdong Lu and Hang Li. 2013. A deep architecture for matching short texts. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023. Fine-tuning LLaMA for multi-stage text retrieval. *ArXiv*, abs/2310.08319.

Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. Cedr: Contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 1101–1104, New York, NY, USA. Association for Computing Machinery.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *ArXiv*, abs/1901.04085.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online. Association for Computational Linguistics.

Cicero Nogueira dos Santos, Xiaofei Ma, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. 2020. Beyond [CLS] through ranking by generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1722–1727, Online. Association for Computational Linguistics.

Liang Pang, Yanyan Lan, J. Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. *ArXiv*, abs/1602.06359.

Ronak Pradeep, Yuqi Liu, Xinyu Zhang, Yilin Li, Andrew Yates, and Jimmy Lin. 2022. Squeezing water from a stone: A bag of tricks for further improving cross-encoder effectiveness for reranking. In *Advances in Information Retrieval*, pages 655–670, Cham. Springer International Publishing.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Devendra Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving passage retrieval with zero-shot question generation. In *Proceedings of* *the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3781–3797, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Seattle, United States. Association for Computational Linguistics.

Richard Socher, Eric Hsin-Chun Huang, Jeffrey Pennington, A. Ng, and Christopher D. Manning. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Neural Information Processing Systems*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. LLAMA 2: Open foundation and fine-tuned chat models. *ArXiv*, arXiv:2307.09288.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*.

Andrew Yates, Siddhant Arora, Xinyu Zhang, Wei Yang, Kevin Martin Jose, and Jimmy Lin. 2020a. Capreolus: A toolkit for end-to-end neural ad hoc retrieval. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, page 861–864, New York, NY, USA. Association for Computing Machinery.

Andrew Yates, Kevin Martin Jose, Xinyu Zhang, and Jimmy Lin. 2020b. Flexible ir pipelines with capreolus. In *Proceedings of the 29th ACM International Conference on Information Knowledge Management*, CIKM '20, page 3181–3188, New York, NY, USA. Association for Computing Machinery.

Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2023. RankT5: Fine-tuning T5 for text ranking with ranking losses. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 2308–2313, New York, NY, USA. Association for Computing Machinery.

A Results on BEIR

Due to the space limitation, we only report the averaged scores on BEIR in the main paper. In this section, Table 5 and Table 6 presents the full nDCG@10 scores on each BEIR dataset, corresponding to the Table 2 in Section 4.1 (**Model Size**), and Table 3 in Section 4.2 (**First-Stage Retriever**).

B License

The MS MARCO dataset is licensed under Creative Commons Attribution 4.0 International, whereas BEIR datasets and Capreolus toolkit are under Apache License 2.0. The usage of the artifacts in this work is consistent with their intended use. Since our codebase is extended from Capreolus, it would inherit the Apache License 2.0.

Backbone	Model	MS MARCO (MRR@10)	BEIR (nDCG@10)													
			Avg	TREC-COVID	NF Corpus	NQ	Hotpot QA	FiQA	Argu Ana	Touche-2020	Quora	DB Pedia	SCI DOCS	FEVER	Climate-FEVER	Sci Fact
MiniLM	monoBERT	0.390	0.467	0.699	0.355	0.504	0.620	0.359	0.335	0.308	0.722	0.426	0.151	0.754	0.164	0.679
	Mean-Pooling	0.391	0.481	0.707	0.351	0.502	0.690	0.356	0.364	0.308	0.807	0.429	0.154	0.721	0.185	0.681
	CELI	0.392	0.491	0.705	0.349	0.501	0.673	0.360	0.527	0.324	0.784	0.424	0.155	0.723	0.172	0.691
ELECTRA$_{base}$	monoBERT	0.400	0.481	0.727	0.362	0.523	0.660	0.389	0.291	0.317	0.773	0.436	0.152	0.748	0.112	0.669
	Mean-Pooling	0.403	0.483	0.732	0.358	0.519	0.718	0.389	0.323	0.321	0.747	0.439	0.149	0.738	0.155	0.689
	CELI	0.402	0.494	0.736	0.368	0.527	0.714	0.401	0.443	0.320	0.690	0.449	0.162	0.740	0.152	0.715
ELECTRA$_{large}$	monoBERT	0.413	0.507	0.801	0.380	0.559	0.733	0.453	0.250	0.339	0.772	0.468	0.181	0.791	0.149	0.719
	Mean-Pooling	0.412	0.516	0.784	0.378	0.554	0.748	0.444	0.332	0.325	0.791	0.456	0.180	0.799	0.215	0.706
	CELI	0.413	0.524	0.786	0.378	0.559	0.735	0.457	0.436	0.335	0.800	0.460	0.182	0.769	0.179	0.733

Table 5: MRR@10 on MS MARCO and nDCG@10 scores on BEIR of CELI and two baselines (i.e., monoBERT and Mean-Pooling). Cross-encoders are initialized from MiniLM, ELECTRA$_{base}$, and ELECTRA$_{large}$. Results on BEIR rerank the top-1k passages from BM25.

First Stage	Model	BEIR (nDCG@10)													
		Avg	TREC-COVID	NF Corpus	NQ	Hotpot QA	FiQA	Argu Ana	Touche-2020	Quora	DB Pedia	SCI DOCS	FEVER	Climate-FEVER	Sci Fact
							Sparse								
BM25	monoBERT	0.467	0.699	0.355	0.504	0.620	0.359	0.335	0.308	0.722	0.426	0.151	0.754	0.164	0.679
	CELI	0.491	0.705	0.349	0.501	0.673	0.360	0.527	0.324	0.784	0.424	0.155	0.723	0.172	0.691
uniCOIL	monoBERT	0.426	0.711	0.337	0.556	0.576	0.271	0.335	0.277	0.727	0.426	0.152	0.375	0.116	0.680
	CELI	0.452	0.713	0.328	0.552	0.625	0.272	0.555	0.285	0.784	0.423	0.156	0.360	0.128	0.691
SPLADE	monoBERT	0.469	0.706	0.336	0.563	0.617	0.362	0.320	0.278	0.728	0.434	0.152	0.758	0.160	0.682
	CELI	0.492	0.699	0.330	0.560	0.671	0.361	0.526	0.288	0.786	0.432	0.157	0.717	0.173	0.691
							Single-vector Dense								
DPR (NQ)	monoBERT	0.451	0.699	0.335	0.571	0.600	0.341	0.333	0.285	0.523	0.433	0.154	0.753	0.175	0.662
	CELI	0.472	0.715	0.330	0.568	0.643	0.339	0.524	0.296	0.557	0.432	0.156	0.721	0.180	0.673
DPR (MS)	monoBERT	0.474	0.737	0.334	0.562	0.613	0.364	0.336	0.278	0.718	0.434	0.153	0.771	0.181	0.677
	CELI	0.495	0.738	0.329	0.557	0.655	0.364	0.528	0.287	0.782	0.434	0.156	0.738	0.186	0.687
ANCE	monoBERT	0.471	0.724	0.331	0.554	0.594	0.360	0.338	0.285	0.717	0.419	0.155	0.781	0.192	0.676
	CELI	0.493	0.740	0.327	0.550	0.626	0.363	0.529	0.291	0.781	0.418	0.157	0.750	0.192	0.687
TCT-ColBERT	monoBERT	0.470	0.719	0.336	0.564	0.620	0.360	0.319	0.281	0.714	0.437	0.154	0.767	0.170	0.676
	CELI	0.494	0.725	0.330	0.560	0.665	0.360	0.524	0.291	0.780	0.438	0.157	0.733	0.177	0.689
TAS-B	monoBERT	0.472	0.714	0.338	0.565	0.623	0.361	0.333	0.281	0.727	0.436	0.153	0.760	0.167	0.680
	CELI	0.494	0.713	0.331	0.560	0.670	0.358	0.527	0.292	0.787	0.435	0.157	0.729	0.176	0.689
							Multi-vector Dense								
ColBERT v2	monoBERT	0.467	0.707	0.333	0.564	0.621	0.360	0.316	0.278	0.716	0.434	0.152	0.756	0.156	0.679
	CELI	0.493	0.709	0.327	0.560	0.672	0.361	0.525	0.291	0.780	0.431	0.157	0.724	0.178	0.691

Table 6: nDCG@10 scores on BEIR, reranking the top-1k passages from each first-stage retriever.