# A Large-Scale Evaluation of Inference-Time Reranker Relevance Feedback

**Anonymous ACL submission**

## Abstract

Neural information retrieval systems often employ a retrieve-and-rerank framework: a bi-encoder retrieves a fixed number of candidates (*e.g.*, $K$=100), which a cross-encoder then reranks. Although the reranker improves the ranking of these candidates, its influence is confined to the top $K$, not affecting Recall@K. Recent studies show that providing relevance feedback from the reranker to the retriever at inference can improve recall. The approach works by updating the retriever's query representations via a distillation process that aligns it with the reranker's predictions. However, existing work have focused on narrow domains like question answering and entity retrieval. In this paper, we study inference-time reranker relevance feedback extensively across multiple retrieval domains, languages, and modalities, while investigating aspects such as the performance and latency implications of the number of distillation updates and feedback candidates.

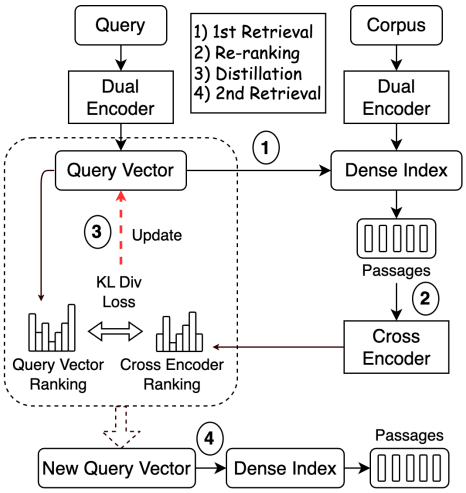


Figure 1: Illustration of REFIT: **Re**ranker Relevance **F**eedback at **I**nference-**T**ime. The distillation process (step 3) updates the query vector in a traditional retrieve-and-rerank framework (steps 1 and 2), improving recall when used for a second retrieval step (step 4).

## 1 Introduction

Retrieve-and-rerank (R&R) (Hu et al., 2019) in neural information retrieval (IR) often employs a *dual-encoder* retriever to retrieve an initial set of candidates in response to a user query (Karpukhin et al., 2020) and a *cross-encoder* reranker to rerank them in order of their relevance (Nogueira and Cho, 2019). While cross-encoders typically offer superior ranking performance, their application is crucially limited to reranking only the top $K$ candidates retrieved by the dual-encoder. Reranking thus does not affect retrieval metrics such as Recall@K, that evaluate the presence of relevant candidates in the top $K$ results. Achieving a high recall, however, can be key in many use cases, especially when retrieved results are used in downstream knowledge-intensive tasks (Petroni et al., 2021).

Some recent studies have proposed improving the recall of retrievers by using relevance feedback from a reranker (Sung et al., 2023; Yadav et al., 2024). The process works by first updating the retriever's query vector via knowledge distillation from the reranker at inference time, which brings the score distributions of the two models over the top $K$ retrieved passages closer. The new query vector is then used to retrieve documents a second time. This approach shares similarities with pseudo-relevance feedback (PRF) (Rocchio, 1971; Lv and Zhai, 2009) in classical IR, which has been used with sparse approaches via query expansion (Bendersky et al., 2011; Xu and Croft, 2017). While these new methods can offer distinct advantages over prior architecture- and language-specific dense PRF methods (Yu et al., 2021; Chandradevan et al., 2022) or training-time reranker distillation (Qu et al., 2021a; Izacard and Grave, 2020; Hofstätter et al., 2020), their efficacy has empirically been tested only on a few narrow use cases such as English phrase and passage retrieval (Sung et al., 2023). This leaves room for further examination of the utility of this promising approach.

This short paper aims to fill that gap by carrying out a large-scale asessment of inference-time reranker feedback into a retriever, which includes multi-domain, multilingual and multimodal evaluation. With our implementation termed REFIT (§2), we demonstrate that reranker relevance feedback generalizes well to a variety of R&R use cases (§3.3). We also present additional analysis of REFIT, including a comparison with training-time reranker distillation (§3.4.1) and the latency-versus-performance tradeoff of different aspects of its underlying distillation mechanism (§3.4.2)

## 2 REFIT: Reranker Relevance Feedback

In this section, we describe how REFIT exploits relevance feedback from a high-accuracy reranker to update the query representations of a retriever during inference (Figure 1). Given a query $q$ and a passage $p$ (or an equivalent unit of retrieval from a non-text modality), let $E_q$ and $E_p$ be their vector representations computed by a bi-encoder retriever $\theta$. A common measure of similarity between $E_q$ and $E_p$, a.k.a. the *retriever score*, is their dot product: $sim(q,p) = E_q^T E_p$. Let $R(q,p)$ be the corresponding *reranker score*.

REFIT performs a lightweight inference-time distillation of $R(q,p)$ scores into $E_q$. More specifically, given retrieved passages $P = \{p_i\}_{i=1}^K$, the reranker $\phi$ first produces a set of scores $R(q,P) = \{R(q,p_i)\}_{i=1}^K$. A reranker scoring distribution is then computed over the elements of $P$:

$$D_\phi(q,P) = \text{softmax}([R(q,p_1),...,R(q,p_K)]) \quad (1)$$

A similar distribution is also computed over the retriever scores:

$$D_\theta(q,P) = \text{softmax}([E_q^T E_{p_1},...,E_q^T E_{p_K}]) \quad (2)$$

REFIT minimizes the following KL-divergence loss during training:

$$\mathcal{L} = D_{KL}(D_\phi(q,P)||D_\theta(q,P)) \quad (3)$$

and updates the query vector via gradient descent. The update process is repeated $n$ times, yielding a new query vector $E_q^{(n)}$, where $n$ is a hyper-parameter. A step-by-step description of the process is given in Algorithm 1, with further details provided in Appendix D.1. The final step in REFIT is to retrieve a second time from the passage index with the new vector $E_q^{(n)}$, with an aim of improving recall over the original vector $E_q$ while retaining the reranking accuracy of the distilled reranker.

**Algorithm 1 REFIT**

**Input**: Query $q$ and its representation $E_q$, retrieved passages $P = [p_1,...,p_K]$ and their representations $[E_{p_1},...,E_{p_K}]$
**Output**: Updated query representation $E_q^{(n)}$

1: Initialize query vector $E_q^{(0)} = E_q$
2: Compute reranker scores $D_\phi(q,P)$ (Eq. 1)
3: **for** $i$ in 0 to $n-1$ **do**
4: Compute retriever scores $D_\theta(q,P)$ with $E_q^{(i)}$ (Eq. 2)
5: Compute loss $\mathcal{L}$ (Eq. 3)
6: Update $E_q^{(i+1)} = E_q^{(i)} - \alpha \frac{\partial}{\partial E_q^{(i)}} \mathcal{L}$
7: **end for**

| Step (Device) | Retrieve & Rerank | | REFIT |
|---|---|---|---|
| | K=100 | K=125 | (K=100) |
| 1st Retrieval (CPU) | 40 | 40 | 40 |
| Rerank (CPU/GPU) | 1540/360 | 1925/450 | 1540/360 |
| Distillation (CPU) | - | - | 30 |
| 2nd Retrieval (CPU) | - | - | 40 |
| Total (CPU/GPU) | 1580/400 | 1965/490 | 1650/470 |

Table 1: Comparison of inference times (in milliseconds) for different approaches, utilizing both CPU-only and GPU configurations, when reranking $K$ passages.

## 3 Experiments

### 3.1 Setup

For a comprehensive assessment of inference-time reranker feedback, we evaluate REFIT in multi-domain, multilingual, and multimodal settings.

**Models:** We use Contriever (Izacard et al., 2021) as our text retriever, available in both English and multilingual versions, and rerankers from sentence transformers (Reimers and Gurevych, 2019). In multimodal evaluation, we use BLIP (Li et al., 2022c), a state-of-the-art vision-language model comprising two unimodal encoders serving as the bi-encoder retriever, and an image-grounded text encoder functioning as the cross-encoder reranker. Further details are provided in Appendix D.2.

**Datasets:** We use BEIR (Thakur et al., 2021) for English multi-domain evaluation, which includes in-domain MS MARCO (Nguyen et al., 2016) as well as out-of-domain test data from various scientific, biomedical, financial, and Wikipedia-based retrieval datasets. For multilingual evaluation, we use two Wikipedia-based benchmarks Mr.TyDi (Clark et al., 2020) and MKQA (Longpre et al., 2021); the former involves retrieval of passages that are in the same language as the query (across 11 languages) while the latter asks to retrieve from the English Wikipedia given queries in 26 different languages. For multimodal evaluation, we use MSRVTT (Xu et al., 2016), which involves text-to-video retrieval.

| | Multi-Domain | | Multilingual | | Multimodal |
|---|---|---|---|---|---|
| | MS Marco | BEIR | Mr.TyDi | MKQA | MSRVTT |
| Retriever | 89.1 | 66.8 | 87.0 | 65.6 | 92.1 |
| Reranker | 89.9 | 67.6 | 88.1 | 66.4 | 92.3 |
| REFIT | **90.5**$^*$ | **69.0**$^*$ | **89.7**$^*$ | **68.2**$^*$ | **92.9**$^+$ |

Table 2: Recall@100 for evaluation in multi-domain, multilingual, and multimodal settings. $^*$ and $^+$ correspond to statistically significant improvements at $p$<0.05 and 0.1, respectively, as per a paired t-test. Full dataset numbers are provided in Tables 8, 9 and 10 in Appendix.

### 3.2 Latency Considerations

REFIT introduces additional overhead to the R&R framework due to distillation and a second retrieval step. Distillation latency scales linearly with the number of updates $n$ (Algorithm 1). Table 1 compares the inference latency of REFIT and standard R&R for text, assuming $n$=100 updates during distillation. The lightweight distillation process takes only 30ms on a CPU. The additional steps increase inference latency for the full pipeline by approximately 4.4% on a CPU and 17.5% on a GPU[1]. In that same amount of time, vanilla R&R can process $K$=125 passages on the GPU (Table 1), to potentially increase Recall@100. Given this observation, and for a fair comparison, we compare REFIT with a *Rerank* baseline that reranks $K$=125 passages instead, while evaluating both on Recall@100.

### 3.3 Main Results

In Table 2 we show the results of evaluating REFIT in English multi-domain, multilingual and multimodal settings. While reranking with $K$=125 expectedly improves Recall@100 over the retriever, REFIT consistently outperforms both. These results underscore the advantage of REFIT's architecture independence and compatibility with arbitrary R&R frameworks. While REFIT primarily aims to improve retriever recall, the ranking performance after the second retrieval step matches that of the reranker, thereby needing no additional reranking after (discussed in Appendix F.4). In addition, Appendix C shows application of REFIT to multi-vector approaches (Santhanam et al., 2022).

### 3.4 Analysis

#### 3.4.1 REFIT vs Training-time Distillation

We compare REFIT's inference-time relevance feedback with RocketQAv2[2] (Ren et al., 2021a), a training-time distillation method. RocketQAv2 builds on top of the RocketQAv1 (Qu et al., 2021b) retriever by leveraging reranker scores for distillation during training. As shown in Table 3, RocketQAv2 outperforms RocketQAv1 on MS MARCO–the dataset used for its distillation–but underperforms on out-of-domain test sets from BEIR. This outcome is not surprising, as training-time distillation restricts the retriever to seeing the reranker's relevance labels only in the source domain(s), limiting generalization to unseen domains. In contrast, REFIT benefits from target-domain pseudolabels provided by the reranker at inference, enhancing out-of-domain generalization.

| | Distillation | MS Marco | BEIR |
|---|---|---|---|
| RocketQAv2 | Training-time | 88.7 | 61.7 |
| RocketQAv1 | None | 88.4 | 63.1 |
| Rerank ($K$=125) | None | 89.4 | 64.4 |
| REFIT ($K$=100) | Inference-time | **90.0** | **65.9** |

Table 3: Recall@100 numbers for comparison against RocketQAv2, a training-time distillation baseline. Rerank and REFIT use RocketQAv1 as the retriever.

#### 3.4.2 Latency vs Performance Tradeoff

Here we investigate the effect of various aspects of REFIT's distillation process on its performance as well as the additional latency they introduce.

**Distillation Updates:** The latency of distillation is linear in the number of updates $n$, with the additional retrieval step taking constant time. In Figure 2 (left), we see that for a mere 4.4% increase in latency ($n$=100), REFIT considerably boosts performance over computationally expensive reranking of $K$=125 candidates (24.3% increase in latency). Under stricter latency constraints, REFIT can be made faster by simply lowering the number of updates, while still surpassing the *Rerank* baseline.

**Passages Reranked:** We investigate how REFIT performs as we vary $K$, the number of passages reranked and subsequently used for distillation. Smaller values of $K$ correspond to a faster R&R pipeline, but it comes at the expense of the target reranker distribution containing less information. In Figure 2 (middle), higher values of $K$ expectedly lead to higher recall in general, while we observe performance improvements over the *Rerank* baseline even with considerably smaller $K$ values.

**Relevance Feedback Iterations:** Finally we examine if further gains are possible from more iterations of relevance feedback, i.e., running the fol-

[1] 24-core AMD EPYC 7352 CPU and 80GB A100 GPU.

[2] As a sanity check, we verified our reproduction of RocketQAv2 by comparing with other works (Table 7 in Appendix).

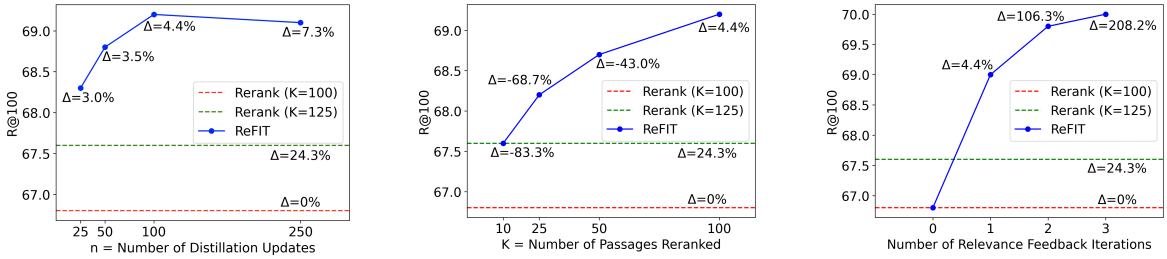


Figure 2: Variation of REFIT performance with distillation updates $n$ (left), reranked passages $K$ used for distillation supervision (middle) and relevance feedback iterations (right). $\Delta$ corresponds to change in latency with respect to the standard R&R framework with $K$=100 (CPU-only configuration).

| Query | Initial Retrieval (within Reranker Top-5) | Newly Retrieved Positive (answer to the query in red) |
|---|---|---|
| who drives the number 95 car in nascar | On October 2013, it was announced that McDowell would be moving to Leavine Family Racing's No. 95 Ford for the 2014 NASCAR Sprint Cup Series season. McDowell failed to qualify for the Daytona 500. | Michael Christopher McDowell is an American professional stock car racing driver. He currently competes full-time in the Monster Energy NASCAR Cup Series, driving the No. 95 Chevrolet SS for Leavine Family Racing. |
| who plays addison shepherd on grey's anatomy | In 2005, she was cast in her breakout role in the ABC series Grey's Anatomy, as Dr. Addison Montgomery, the estranged spouse of Derek Shepherd. | Kathleen Erin Walsh is an American actress and businesswoman. Her roles include Dr. Addison Montgomery on the ABC television dramas Grey's Anatomy and Private Practice. |

Table 4: Examples of how initial retrievals highly ranked (top-5) by the reranker (middle) helps retrieve new positives (right) via the updated query vector, due to important lexical and semantic overlap (highlighted in green).

lowing steps in a loop: (1) rerank the retrieved documents from the previous iteration, (2) update the query vector via distillation from the reranker distribution, and (3) retrieve again. In Figure 2 (right), we see that recall does go up with more rounds of relevance feedback; the biggest gain, however, comes in the first round, while later rounds give diminishing returns along with a considerable increase in latency from needing to perform the computationally expensive reranking at every iteration.

### 3.4.3 Comparison with TOUR

We also compare REFIT with TOUR (Sung et al., 2023), an existing inference-time reranker feedback mechanism, on the passage retrieval benchmarks used by Sung et al. (2023) as well as the multidomain BEIR and multilingual Mr.TyDi benchmarks. For NQ and EntityQuestions, we employ the same retriever (Karpukhin et al., 2020) and reranker (Fajcik et al., 2021) as in Sung et al. (2023), while for BEIR and Mr.TyDi, we use the models described in §3.1. As shown in Table 5, REFIT consistently outperforms both TOUR variants. Please see Appendix B for a detailed discussion on TOUR.

### 3.5 Discussion

REFIT adjusts the query vector using reranker feedback to align it with passages deemed more relevant. Across different BEIR datasets, this adjusted query vector is 5–16% closer to the initially retrieved positives. New positives identified by the updated query vector are closest to a passage in the reranker's top 5 in 26% of cases (38% for top 10; 55% for top 20), indicating effective knowledge transfer from the reranker. Table 4 provides examples where specific words and phrases from a passage in the reranker's top 5 help retrieve additional candidates with lexical or semantic overlap (highlighted in green) via relevance feedback. Notably, in the second example, an incorrect passage highly ranked by the reranker leads to the retrieval of a correct positive. More details in Appendix F.

| | NQ | EntityQ | BEIR | Mr.TyDi |
|---|---|---|---|---|
| Retriever | 86.1 | 70.1 | 66.8 | 87.0 |
| Reranker ($K$=125) | 86.8 | 71.2 | 67.6 | 88.1 |
| TOUR$_{\text{hard}}$ ($K$=100) | 87.0$^+$ | 71.9$^+$ | 68.4 | 88.7 |
| TOUR$_{\text{soft}}$ ($K$=100) | 87.2$^+$ | 72.5$^+$ | 68.1 | 88.1 |
| REFIT ($K$=100) | **87.6** | **72.6** | **69.0***| **89.7*** |

Table 5: Recall@100 numbers for comparison of REFIT with both variants of TOUR. $^+$ corresponds to numbers directly taken from Sung et al. (2023). Improvements with * are significant at $p < 0.05$ as per paired t-test.

## 4 Conclusion

We extensively evaluate REFIT, our implementation of inference-time reranker relevance feedback, and demonstrate its effectiveness in multi-domain, multilingual and multimodal settings. We show that REFIT outperforms training-time distillation, extends to even multi-vector approaches, while introducing considerably low additional latency into the R&R framework. Future work will focus on leveraging textual relevance feedback from LLMs.

## Limitations

REFIT introduces an additional latency into a traditional retrieve-and-rerank framework. The distillation time is only dependent on the number of updates, and is unaffected by the model architecture and number of retrieved passages; the overall additional latency (30ms when $n$=100 for distillation plus 40ms for the second retrieval) amounts to an extra 17.5% on GPU (or 4.4% on CPU) when the number of retrieved passages $K$=100. However, it is noteworthy that REFIT remains faster and exhibits superior performance compared to the standard approach of reranking a larger pool of candidates for improving recall. Moreover, the efficacy of our approach is contingent upon the reranker providing a better ranking than the retriever. We anticipate that our method might provide minimal gains in situations where the performance of the retriever is close to that of the reranker.

## References


Michael Bendersky, Donald Metzler, and W Bruce Croft. 2011. Parameterized concept weighting in verbose queries. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 605–614.

Ramraj Chandradevan, Eugene Yang, Mahsa Yarmohammadi, and Eugene Agichtein. 2022. Learning to enrich query representation with pseudo-relevance feedback for cross-lingual retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1790–1795.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Martin Fajcik, Martin Docekal, Karel Ondrej, and Pavel Smrz. 2021. R2-D2: A modular baseline for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 854–870, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving efficient neural ranking models with cross-architecture knowledge distillation. *arXiv preprint arXiv:2010.02666*.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. 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*, pages 113–122.

Minghao Hu, Yuxing Peng, Zhen Huang, and Dongsheng Li. 2019. Retrieve, read, rerank: Towards end-to-end multi-document reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2285–2295.

Gautier Izacard and Edouard Grave. 2020. Distilling knowledge from reader to retriever for question answering. In *International Conference on Learning Representations*.

Gautier Izacard, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

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 International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Canjia Li, Yingfei Sun, Ben He, Le Wang, Kai Hui, Andrew Yates, Le Sun, and Jungang Xu. 2018. Nprf: A neural pseudo relevance feedback framework for ad-hoc information retrieval. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4482–4491.

Hang Li, Ahmed Mourad, Bevan Koopman, and Guido Zuccon. 2022a. How does feedback signal quality impact effectiveness of pseudo relevance feedback for passage retrieval? *arXiv preprint arXiv:2205.05888*.

Hang Li, Shengyao Zhuang, Ahmed Mourad, Xueguang Ma, Jimmy Lin, and Guido Zuccon. 2022b. Improving query representations for dense retrieval with pseudo relevance feedback: A reproducibility study. In *European Conference on Information Retrieval*, pages 599–612.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022c. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. *arXiv preprint arXiv:2302.07452*.

Shayne Longpre, Yi Lu, and Joachim Daiber. 2021. Mkqa: A linguistically diverse benchmark for multilingual open domain question answering. *Transactions of the Association for Computational Linguistics*, 9:1389–1406.

Yuanhua Lv and ChengXiang Zhai. 2009. Adaptive relevance feedback in information retrieval. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 255–264.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPs*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. 2021. Kilt: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021a. 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*, pages 5835–5847.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021b. 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*, pages 5835–5847, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021a. RocketQAv2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2825–2835, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021b. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2825–2835.

Joseph Rocchio. 1971. Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing*, pages 313–323.

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*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Mujeen Sung, Jungsoo Park, Jaewoo Kang, Danqi Chen, and Jinhyuk Lee. 2023. Optimizing test-time query representations for dense retrieval. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5731–5746, Toronto, Canada. Association for Computational Linguistics.

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)*.

Yanmeng Wang, Jun Bai, Ye Wang, Jianfei Zhang, Wenge Rong, Zongcheng Ji, Shaojun Wang, and Jing Xiao. 2021. Enhancing dual-encoders with question and answer cross-embeddings for answer retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2306–2315.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Jinxi Xu and W Bruce Croft. 2017. Quary expansion using local and global document analysis. In *Acm sigir forum*, volume 51, pages 168–175. ACM New York, NY, USA.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296.

Nishant Yadav, Nicholas Monath, Manzil Zaheer, Rob Fergus, and Andrew McCallum. 2024. Adaptive retrieval and scalable indexing for k-nn search with cross-encoders. In *The Twelfth International Conference on Learning Representations*.

HongChien Yu, Chenyan Xiong, and Jamie Callan. 2021. Improving query representations for dense retrieval with pseudo relevance feedback. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3592–3596.

Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, and Andrew Yates. 2020. Bert-qe: Contextualized query expansion for document re-ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4718–4728.

## A Related Work

**Pseudo-relevance feedback:** REFIT has similarities with Pseudo-Relevance Feedback (PRF) (Rocchio, 1971; Lv and Zhai, 2009; Li et al., 2022a) in IR: Bendersky et al. (2011); Xu and Croft (2017) use the retrieved documents to improve sparse approaches via query expansion or query term reweighting, Li et al. (2018); Zheng et al. (2020) score similarity between a target document and a top-ranked feedback document, while Yu et al. (2021) train a separate query encoder that computes a new query embedding using the retrieved documents as additional input. In contrast, REFIT does not require customized training feedback models or the availability of explicit feedback data, as we improve the query vector by directly distilling from the reranker's output within an R&R framework.

Further, previous approaches to PRF have been dependent on the choice of retriever architecture and language; Yu et al. (2021)'s PRF model is tied to the retriever used, Chandradevan et al. (2022) explore cross-lingual relevance feedback, but require feedback documents in the target language and thereby could only apply to three languages. In contrast, REFIT is independent of the choice of R&R architecture and can be used for neural retrieval in any domain, language, or modality.

**Distillation in Neural IR:** Existing approaches primarily leverage reranker feedback *during training* of the dual-encoder retriever, to sample better negatives (Qu et al., 2021a), for standard knowledge distillation of the cross-attention scores (Izacard and Grave, 2020), to train smaller and more efficient rankers by distilling larger models (Hofstätter et al., 2020), or to align the geometry of dual-encoder embeddings with that from cross-encoders (Wang et al., 2021). Instead, REFIT leverages distillation at inference time, updating only the query representation to replicate the cross-encoder's scores for the corresponding test instance. A key implication of this design choice is that, unlike existing methods, the retriever parameters are kept unchanged, meaning REFIT can be incorporated out-of-the-box into any neural R&R framework. In contrast, extending training-time distillation to new languages or modalities would require re-training the bi-encoder.

## B Discussion on comparison with TOUR

TOUR (Sung et al., 2023) proposes test-time optimization of query representations with two variants: $\text{TOUR}_{\text{hard}}$ and $\text{TOUR}_{\text{soft}}$. $\text{TOUR}_{\text{hard}}$ optimizes the marginal likelihood of a small set of (pseudo) positive contexts. REFIT shares similarities with $\text{TOUR}_{\text{soft}}$, which uses the normalized scores of a cross-encoder over the retrieved results as soft labels. TOUR was evaluated only on English phrase and passage retrieval tasks, while we demonstrate REFIT's effectiveness in multidomain, multilingual and multimodal settings. From the results in Table 5, we attribute TOUR's lower performance to its early stopping criterion for distillation updates; specifically, TOUR conducts relevance feedback for three iterations, continuing distillation in each until the top-1 retrieval has the highest reranker score (for $\text{TOUR}_{\text{soft}}$) or is a pseudo-positive (for $\text{TOUR}_{\text{hard}}$). In contrast, REFIT performs more distillation updates within a single iteration (tuned to $n$=100 updates), making it considerably faster, since each additional iteration of relevance feedback incurs a significant computational overhead.

## C REFIT with Multi-Vector Approaches

While our experiments thus far have focused on single-vector dense retrieval, encoding queries and passages as individual vectors, multi-vector retrieval models (Khattab and Zaharia, 2020; Santhanam et al., 2022) compute token-level representations with late interaction scoring. We now explore applying REFIT to multi-vector retrieval, specifically ColBERTv2 (Santhanam et al., 2022), where the distillation (Step 3 in Figure 1) updates individual token embeddings in the query. Table 6 presents results on a subset[3] of the BEIR dataset, demonstrating that REFIT effectively extends to multi-vector dense retrieval. Notably, since ColBERTv2 was trained using reranker scores for supervision, our results reinforce the finding from §3.4.1 that inference-time distillation can outperform standard distillation during training.

| | ColBERTv2 | Reranker | REFIT |
|---|---|---|---|
| NFCorpus | 27.7 | 28.0 | **28.8** |
| FiQA | 62.8 | 63.7 | **64.3** |
| Scidocs | 35.8 | 36.6 | **38.5** |
| Scifact | 89.4 | **90.2** | 90.1 |

Table 6: Recall@100 on a subset of BEIR with ColBERTv2 as underlying retriever for Rerank and REFIT.

[3] Due to the substantially larger index size for multi-vector retrieval approaches, we restrict to experiments here to BEIR subsets with $<$ 100k passages in the retrieval corpus.

## D Experimental Details

### D.1 Distillation

We observe that the output scores from the dual-encoder and the cross-encoder models are not bounded to specific intervals. Hence, we do min-max normalization separately on the query vector's scores $Q_q^T \hat{P}$ (from the dual-encoder) and the cross encoder's scores $R(q, P)$ to bring the two scoring distributions closer. Further, the cross-encoder tends to have peaky scoring distributions, hence we use a temperature $T$ (= 2 after hyperparameter-tuning) while computing the softmax $D_{CE}(q, P)$ over the cross-encoder scores. After tuning on the MS Marco dev set, we set the number of updates $n$=100 with learning rate $\alpha$=0.005.

### D.2 Retriever and Reranker

We use Contriever (Izacard et al., 2021) as the underlying text retriever, which has been pre-trained with an unsupervised contrastive learning objective on a large-scale collection of Wikipedia and CCNet documents. Contriever is a SOTA dual-encoder retriever that outperforms traditional term-matching methods, BM25 and recent dense approaches e.g. DPR (Karpukhin et al., 2020), ANCE (Xiong et al., 2020) and ColBERT (Khattab and Zaharia, 2020). For retrieval in both English and other languages, we use the publicly available version of Contriever, fine-tuned on MS MARCO (Nguyen et al., 2016). Our English[4] and multilingual[5] rerankers are based on sentence transformers.

The retriever and reranker for multimodal settings are based on BLIP (Li et al., 2022c), a state-of-the-art vision-language model that comprises two unimodal encoders and an image-grounded text encoder. The unimodal encoders encode image and text separately, akin to dual-encoders in text-to-text retrieval, and are trained with an Image-Text Contrastive (ITC) loss. The image-grounded text encoder injects visual information into the text encoder by incorporating a cross-attention layer, similar to a text-to-text cross-encoder, and is trained with an Image-Text Matching (ITM) loss. We refer the reader to (Li et al., 2022c) for a more detailed description of BLIP's architecture and the pre-training objectives. BLIP can thus be used for retrieval with the unimodal encoders (which we refer to as BLIP$_{ITC}$), and for reranking with the image-grounded text encoder (which we refer to as BLIP$_{ITM}$). We use the output from BLIP$_{ITM}$ as the reranker distribution, which is then used to compute the distillation loss for updating the query representation that is output by BLIP$_{ITC}$. The *Rerank* baseline uses BLIP$_{ITM}$ to rerank $K$=125 videos retrieved by BLIP$_{ITC}$.

[4] cross-encoder/ms-marco-MiniLM-L-6-v2

[5] cross-encoder/mmarco-mMiniLMv2-L12-H384-v1

## E Experiments with PRF baselines

We also evaluated ANCE-PRF (Yu et al., 2021) as a Pseudo-Relevance Feedback (PRF) baseline. ANCE-PRF trains a new PRF query encoder to output the query embedding by taking the query and top K (=5) documents retrieved by ANCE (concatenated with separator tokens) as input. In our experiments, we trained a PRF query encoder that takes the top K documents from our baseline retriever model, Contriever, as input. However, we found that the query embedding produced by the new PRF encoder did not improve upon the baseline Contriever model. This finding aligns with a previous study (Li et al., 2022b) that investigated the applicability of the ANCE-PRF method using dense retrievers other than ANCE. According to Li et al. (2022b), improvement from the PRF query encoder was minimal once dense retrievers stronger than ANCE were considered. Further, Li et al. (2022b) considered dense retrievers (Hofstätter et al., 2021) which were even weaker compared to the more powerful and recent Contriever model that we use as our retriever.

## F Further Discussion

### F.1 Query Vectors: The Original & The New

To better understand how the updated query vector after reranker relevance feedback improves recall, we take a closer look at the query and passage vectors computed for a set of BEIR examples. Figure 3 shows t-SNE plots for four such examples, where each dot represents a vector, and the distance between any two points is their cosine distance. As the figure clearly suggests, the reranker feedback brings the query vector in each case closer to the corresponding positive passage vectors, making the query align with an increased number of relevant passages and consequently improving recall.

### F.2 The Curious Case of Zero Initial Positives

In §F.1, we presented an example where REFIT leverages a close negative among the initially retrieved candidates to later retrieve a positive passage. We find that in 24% of the cases where the

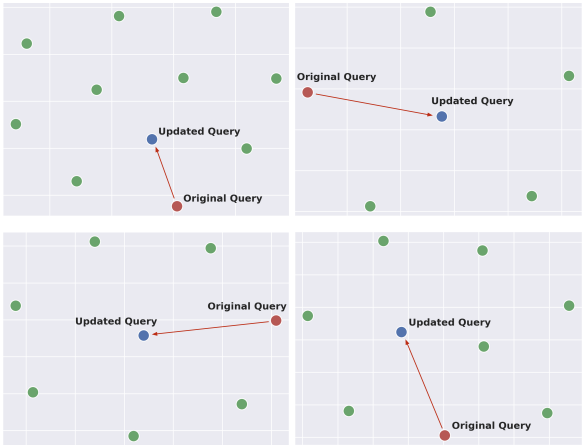


Figure 3: t-SNE plots for examples from BEIR, with the query vectors shown alongside the corresponding positive passages. The updated query vectors after REFIT are now closer to the positive passages (in green).

first-stage retriever retrieves no positive passages, REFIT can improve recall in a similar fashion. Among all cases where recall improves, however, 75% have at least one positive in the top retrieved results. These results indicate that while the presence of positive candidates in the initial retrieval is useful, reranker relevance feedback approach can also generally leverage informative negatives to update the query vector in the right direction.

### F.3 Choice of Reranker

In the experiments comparing REFIT to the R&R framework, we used an efficient (yet high-performing) reranker both in the baseline model and as the teacher model for distillation. Would the results have been different if we used a more powerful (but computationally expensive) reranker instead? To find an answer, it is essential to note that the final recall of an R&R engine is inherently limited by the underlying retriever. For instance, the Recall@100 of an R&R pipeline with $K$=125 cannot exceed the Recall@125 of the underlying retriever, irrespective of the quality of the reranker. The Recall@100 of REFIT (BEIR: 69.0, Mr. TyDi: 89.7, MKQA: 68.2 and MSRVTT: 92.9) is consistently higher than the Recall@125 of the baseline retriever (BEIR: 68.9, Mr. TyDi: 88.2, MKQA: 66.9 and MSRVTT: 92.8). These results clearly suggest that even the best reranker baseline would fail to attain the recall of REFIT. Further, we can safely expect a better reranker to improve the recall of REFIT since leveraging a stronger teacher model (reranker) for distillation should lead to a better student (retriever query vector).

### F.4 Recall Improves, but How Good is the Ranking?

In §3.3, we demonstrated that REFIT improves Recall@100 over existing retrieval-only and reranking baselines. In many applications including web search, the ranks of the retrieved results matter too. One important question for our method, therefore, is whether its output scores translate to an accurate ranking of the retrieved results; a failure to do so may necessitate a second expensive call to the reranker. We evaluate REFIT for ranking on BEIR, Mr. TyDi and MKQA using the respective official ranking metrics. We note that MKQA contains judgments in the form of answer spans, and prior work only reports recall-based metrics. Hence, we use Recall@20 as a pseudo-measure of ranking quality in case of MKQA. Table 11 shows the results: while Rerank expectedly yields big gains over Contriever, REFIT is able to match Rerank on ranking performance. Combined with its improved retrieval recall, this indicates that REFIT is suitable for both retrieval and ranking-oriented applications.

| Domain | Published | Our Reproduction |
|---|---|---|
| Trec-Covid | 67.5 | 68.6 |
| NFCorpus | 29.3 | 29.4 |
| NQ | 50.5 | 50.2 |
| HotpotQA | 53.3 | 53.6 |
| FiQA | 30.2 | 30.4 |
| DBPedia | 35.6 | 35.8 |
| Scidocs | 13.1 | 12.8 |
| FEVER | 67.6 | 69.0 |
| Climate-FEVER | 18.0 | 19.1 |
| Scifact | 56.8 | 55.1 |
| Average | 42.2 | 42.4 |

Table 7: nDCG@10 scores from our reproduction of RocketQAv2 compared those in ColBERTV2 (Santhanam et al., 2022) and DRAGON (Lin et al., 2023).

| | mBERT | XLM-R | Contriever | Rerank | REFIT |
|---|---|---|---|---|---|
| Arabic | 81.1 | 79.9 | 88.7 | 89.5 | **90.9*** |
| Bengali | 88.7 | 84.2 | 91.4 | 91.4 | **95.9*** |
| English | 77.8 | 73.1 | 77.2 | 78.7 | **81.8*** |
| Finnish | 74.2 | 81.6 | 88.1 | 88.9 | **91.0*** |
| Indonesian | 81.0 | 87.4 | 89.8 | 90.5 | **93.7*** |
| Japanese | 76.1 | 70.9 | 81.7 | 82.5 | **85.2*** |
| Korean | 66.7 | 71.1 | 78.2 | **81.0** | 80.2 |
| Russian | 77.6 | 74.1 | 83.8 | 85.7 | **87.3** |
| Swahili | 74.1 | 73.9 | 91.4 | **92.0** | 90.5 |
| Telugu | 89.5 | 91.2 | 96.6 | 97.0 | **97.5** |
| Thai | 57.8 | 89.5 | 90.5 | 91.6 | **93.3*** |
| Average | 76.8 | 79.7 | 87.0 | 88.1 | **89.7*** |

Table 8: Recall@100 (in %) on the multilingual Mr.TyDi benchmark. Rerank and REFIT use Contriever as the underlying Retriever. Improvements with * are significant at $p < 0.05$ according to the paired t-test.

| | BM25 | ANCE | RocketQAv1 | RocketQAv2 | RocketQAv1 + Rerank | RocketQAv1 + REFIT | Contriever | Contriever + Rerank | Contriever + REFIT |
|---|---|---|---|---|---|---|---|---|---|
| MS MARCO | 65.8 | 85.2 | 88.4 | 88.7 | 89.4 | **90.0*** | 89.1 | 89.9 | **90.5*** |
| Trec-COVID | 49.8 | 45.7 | 48.5 | 46.4 | 52.0 | **52.9** | 40.7 | 43.8 | **51.5*** |
| NFCorpus | 25.0 | 23.2 | 26.9 | 25.9 | 27.4 | **29.2*** | 30.0 | 29.5 | **31.9*** |
| NQ | 76.0 | 83.6 | 91.1 | 89.8 | 91.8 | **92.7*** | 92.5 | 93.3 | **94.2*** |
| HotpotQA | 74.0 | 57.8 | 69.8 | 67.7 | 71.4 | **73.3*** | 77.7 | 78.6 | **80.4*** |
| FiQA | 53.9 | 58.1 | 63.6 | 61.2 | **64.3** | 63.8 | 65.6 | **65.9** | 65.6 |
| DBPedia | 39.8 | 31.9 | 45.7 | 43.4 | 47.6 | **50.2*** | 54.1 | 56.0 | **57.3*** |
| Scidocs | 35.6 | 26.9 | 31.8 | 29.3 | 33.1 | **35.5*** | 37.8 | 38.3 | **40.1*** |
| FEVER | 93.1 | 90.0 | 92.6 | 92.5 | 92.8 | **93.7*** | 94.9 | 95.3 | **95.5*** |
| Climate-FEVER | 43.6 | 44.5 | 47.4 | 48.7 | 49.3 | **53.6*** | 57.4 | 59.0 | **59.5** |
| Scifact | 90.8 | 81.6 | 88.1 | 85.4 | 89.0 | **89.9*** | 94.7 | 94.4 | **95.2*** |
| Average | 58.9 | 57.1 | 63.1 | 61.7 | 64.4 | **65.9*** | 66.8 | 67.6 | **69.0*** |

Table 9: Recall@100 (in %) on the English BEIR benchmark. Performance of REFIT is shown for different choices of underlying retrievers. RocketQAv2 (Ren et al., 2021b) corresponds to a training-time distillation baseline. Improvements marked with * are statistically significant at $p < 0.05$ as per paired t-test.

| | avg | en | ar | fi | ja | ko | ru | es | sv | he | th | da | de | fr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| mBERT | 57.9 | 74.2 | 44.0 | 51.7 | 55.7 | 48.2 | 57.4 | 63.9 | 62.7 | 46.8 | 51.7 | 63.7 | 59.6 | 65.2 |
| XLM-R | 59.2 | 73.4 | 42.4 | 57.7 | 53.1 | 48.6 | 58.5 | 62.9 | 67.5 | 46.9 | 61.5 | 66.9 | 60.9 | 62.4 |
| Contriever | 65.6 | 75.6 | 53.3 | 66.6 | 60.4 | 55.4 | 64.7 | 70.0 | 70.8 | 59.6 | 63.5 | 72.0 | 66.6 | 70.1 |
| Rerank | 66.4 | 76.0 | 54.5 | 67.5 | 61.5 | 56.7 | 65.8 | 70.5 | 71.6 | 60.8 | 64.9 | 72.7 | 67.5 | 70.6 |
| REFIT | **68.2** | **76.6** | **58.0** | **68.8** | **64.7** | **59.3** | **68.4** | **72.5** | **73.1** | **62.9** | **66.5** | **74.1** | **70.1** | **72.5** |

| | it | nl | pl | pt | hu | vi | ms | km | no | tr | cn | hk | tw |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| mBERT | 64.1 | 66.7 | 59.0 | 61.9 | 57.5 | 58.6 | 62.8 | 32.9 | 63.2 | 56.0 | 58.4 | 59.3 | 59.3 |
| XLM-R | 58.1 | 66.4 | 61.0 | 62.0 | 60.1 | 62.4 | 66.1 | **46.6** | 65.9 | 60.6 | 55.8 | 55.5 | 55.7 |
| Contriever | 70.3 | 71.4 | 68.8 | 68.5 | 66.7 | 67.8 | 71.6 | 37.8 | 71.5 | 68.7 | 64.1 | 64.5 | 64.3 |
| Rerank | 70.8 | 72.0 | 69.9 | 69.3 | 67.5 | 68.7 | 72.0 | 38.6 | 72.3 | 69.3 | 65.1 | 65.4 | 65.2 |
| REFIT | **72.4** | **73.6** | **71.1** | **71.5** | **68.9** | **70.5** | **73.3** | 39.9 | **73.3** | **70.7** | **67.5** | **67.4** | **66.9** |

Table 10: Recall@100 (in %) on the cross-lingual MKQA benchmark. Rerank and REFIT use Contriever as the underlying retriever. All improvements are statistically significant at $p < 0.05$ as per paired t-test.

| | BM25 | ANCE | Contriever | Rerank | REFIT |
|---|---|---|---|---|---|
| MS MARCO | 22.8 | 38.8 | 40.7 | 47.0 | 47.0 |
| Trec-COVID | 65.6 | 65.4 | 59.6 | 71.0 | 72.3 |
| NFCorpus | 32.5 | 23.7 | 32.8 | 34.5 | 35.1 |
| NQ | 32.9 | 44.6 | 49.8 | 57.6 | 57.7 |
| HotpotQA | 60.3 | 45.6 | 63.8 | 71.8 | 71.6 |
| FiQA | 23.6 | 29.5 | 32.9 | 35.6 | 36.7 |
| DBPedia | 31.3 | 28.1 | 41.3 | 47.0 | 47.1 |
| Scidocs | 15.8 | 12.2 | 16.5 | 17.0 | 17.1 |
| FEVER | 75.3 | 66.9 | 75.8 | 81.9 | 81.9 |
| Cli.-FEVER | 21.3 | 19.8 | 23.7 | 25.5 | 25.7 |
| Scifact | 66.5 | 50.7 | 67.7 | 69.1 | 69.3 |
| Average | 40.7 | 38.7 | 45.9 | 50.7 | 51.0 |

(a) nDCG@10 on the English BEIR benchmark.

| | mBERT | XLM-R | Contriever | Rerank | REFIT |
|---|---|---|---|---|---|
| Arabic | 34.8 | 36.5 | 43.4 | 63.1 | 62.9 |
| Bengali | 35.1 | 41.7 | 42.3 | 62.4 | 62.9 |
| English | 25.7 | 23.0 | 27.1 | 37.7 | 37.9 |
| Finnish | 29.6 | 32.7 | 35.1 | 50.0 | 50.2 |
| Indonesian | 36.3 | 39.2 | 42.6 | 60.8 | 61.0 |
| Japanese | 27.1 | 24.8 | 32.4 | 51.0 | 50.9 |
| Korean | 28.1 | 32.2 | 34.2 | 49.2 | 48.8 |
| Russian | 30.0 | 29.3 | 36.1 | 55.1 | 55.0 |
| Swahili | 37.4 | 35.1 | 51.2 | 64.0 | 64.1 |
| Telugu | 39.6 | 54.7 | 37.4 | 76.1 | 76.1 |
| Thai | 20.3 | 38.5 | 40.2 | 67.1 | 67.0 |
| Average | 31.3 | 35.2 | 38.4 | 57.9 | 57.9 |

(b) MRR@100 on the multilingual Mr.TyDi benchmark.

| | avg | en | ar | fi | ja | ko | ru | es | sv | he | th | da | de | fr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| mBERT | 45.3 | 65.5 | 30.2 | 38.9 | 41.7 | 34.5 | 44.3 | 52.4 | 50.5 | 32.6 | 38.5 | 52.5 | 46.6 | 53.8 |
| XLM-R | 46.7 | 64.5 | 29.0 | 45.1 | 39.7 | 34.9 | 45.9 | 51.4 | 56.1 | 32.5 | 49.4 | 55.8 | 48.3 | 50.5 |
| Contriever | 53.9 | 67.2 | 40.1 | 55.1 | 46.2 | 41.7 | 52.3 | 59.3 | 60.0 | 45.6 | 52.0 | 62.0 | 54.8 | 59.3 |
| Rerank | 59.6 | 70.7 | 47.0 | 59.9 | 54.6 | 49.1 | 59.3 | 64.7 | 65.5 | 53.2 | 57.9 | 66.1 | 61.5 | 64.3 |
| REFIT | 59.6 | 70.6 | 46.9 | 59.8 | 54.7 | 49.2 | 59.3 | 64.9 | 65.4 | 53.3 | 57.9 | 65.8 | 61.7 | 64.5 |

| | it | nl | pl | pt | hu | vi | ms | km | no | tr | cn | hk | tw |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| mBERT | 52.1 | 55.3 | 45.6 | 49.5 | 44.6 | 46.9 | 49.9 | 21.5 | 51.3 | 42.7 | 44.6 | 45.3 | 45.5 |
| XLM-R | 45.4 | 54.5 | 48.5 | 49.6 | 47.3 | 49.7 | 54.0 | 33.4 | 53.7 | 48.7 | 42.4 | 42.4 | 42.0 |
| Contriever | 59.4 | 60.9 | 58.1 | 56.9 | 55.2 | 55.9 | 60.9 | 26.2 | 61.0 | 56.7 | 50.9 | 51.9 | 51.2 |
| Rerank | 64.9 | 66.1 | 62.8 | 63.0 | 59.9 | 62.6 | 65.4 | 30.5 | 65.9 | 61.9 | 57.8 | 59.2 | 57.5 |
| REFIT | 64.8 | 66.0 | 63.0 | 63.4 | 59.7 | 62.3 | 65.3 | 30.7 | 65.3 | 62.1 | 60.0 | 58.3 | 57.5 |

(c) Recall@20 as a pseudo-measure of ranking for different approaches on the cross-lingual MKQA benchmark.

Table 11: Comparison of the ranking performance on the BEIR, Mr. TyDi and MKQA benchmarks.