# HYRR: Hybrid Infused Reranking for Passage Retrieval

**Jing Lu***, **Keith Hall**†, **Ji Ma*** **and Jianmo Ni**◇
*Google Research, †Sizzle AI, ◇Google DeepMind
{ljwinnie, maji, jianmon}@google.com, khallbobo@gmail.com

**Abstract**

Existing passage retrieval systems typically adopt a two-stage retrieve-then-rerank pipeline. To obtain an effective reranking model, many prior works have focused on improving the model architectures, such as leveraging powerful pretrained large language models (LLM) and designing better objective functions. However, less attention has been paid to the issue of collecting high-quality training data. In this paper, we propose HYRR, a framework for training robust reranking models. Specifically, we propose a simple but effective approach to select training data using hybrid retrievers. Our experiments show that the rerankers trained with HYRR are robust to different first-stage retrievers. Moreover, evaluations using MS MARCO and BEIR data sets demonstrate our proposed framework effectively generalizes to both supervised and zero-shot retrieval settings.

**Keywords:** passage retrieval, text ranking

## 1. Introduction

Recent passage retrieval systems have generally seen pipelined retrieve-then-rerank approaches achieve the best performance. The first stage utilizes an efficient retrieval model that retrieves a set of candidate passages for a given query from the entire corpus. Subsequently, the second stage employs a slower but more effective reranking model that reranks the candidates to produce the final ranking. Significant progress has been made recently on neural retrieval models (Karpukhin et al., 2020; Qu et al., 2021; Ren et al., 2021; Ni et al., 2022) by leveraging pretrained large language models (LLM), such as BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020). Also, powerful neural reranking models have been proposed through fine-tuning LLMs (Nogueira and Cho, 2019; Nogueira et al., 2020; Pradeep et al., 2021; Zhuang et al., 2023b) or prompting LLMs (Sachan et al., 2022; Liang et al., 2023; Qin et al., 2023; Sun et al., 2023; Ma et al., 2023; Zhuang et al., 2023a).

Some prior works learn rerankers independently of the first-stage retrievers (Nogueira and Cho, 2019; Nogueira et al., 2019, 2020; Pradeep et al., 2021). For example, monoT5 (Nogueira et al., 2020) is trained using labeled examples in the MS MARCO (Nguyen et al., 2016), a dataset sampled from real and anonymized Bing queries and then is applied on the candidates retrieved by a BM25 retriever. The relevant and non-relevant passages in MS MARCO are collected from the top search results from BING search engine, which may not reflect the distribution of top retrieved results of the BM25 retriever. In addition, as pointed out by Gao et al. (2021), when the retrieval model used to generate negative training examples is weaker than the test retrieval model to which the reranking model applies, the performance of the reranker drops severely. Some other works train rerankers on data that is similar to the distribution that the reranker will observe at inference time. That is, the first-stage retriever produces candidates for the reranker for both training and inference (Huang et al., 2020; Ren et al., 2021; Bonifacio et al., 2022; Zhuang et al., 2023b). While we do see this approach achieving strong results in general, we need to retrain the reranker when the first-stage retriever is changed. Previous studies have demonstrated that there is a trade-off between the training complexity and model performance for reranking models. Rerankers that are trained independently of the first-stage retriever can be easier to train but may not achieve optimal performance. In contrast, rerankers that are trained in a dependent manner on the first-stage retriever can achieve superior performance but are more complex to train.

In this work, we revisit the question of how best to train a reranking model for retrieval. We show that by training a robust reranker which has been exposed to training data from a hybrid of term-based and neural retrieval models, we are able to achieve strong performance no matter what retrieval model is used in the first stage. Specifically, the retrieval model we used to generate training candidates for rerankers is a hybrid retriever, inspired by Ma et al.'s (2021) hybrid first-stage retriever. For each passage (and query), the encoding from a term-based sparse retrieval model and the encoding from a neural retrieval model are concatenated to form a hybrid encoding. We then perform approximate nearest neighbor search over these hybrid encodings. This results in a different candidate set than independently selecting neighbors from the sparse retriever and the neural retriever. We then sample negative examples from the top retrieved



*LREC-COLING 2024*, pages 8528–8534

results of this hybrid retriever.

Experiments on MS MARCO passage ranking task and BEIR (Thakur et al., 2021) retrieval tasks show that our proposed training framework is effective in both supervised setting and zero-shot setting. We show that this approach results in a robust reranker which performs well across different retrievers, domains, and tasks.

The primary contributions of this paper are:

- We present **HYRR**, a training paradigm for training rerankers based on hybrid term-based and neural retrievers.
- We show that this approach is effective in both supervised setting and zero-shot setting.
- We show that this approach results in a robust reranker which performs well across different retrievers, domains, and tasks; though there are still limitations which appear to be based in the query generation approach utilized in the zero-shot setting.

## 2. Related Work

A number of prior works have explored using neural models for text ranking, with recent focus on transformer-based models (Vaswani et al., 2017). Even through it is computationally expensive, a BERT-based cross-attention model is one of the most dominant models for text ranking (Nogueira and Cho, 2019; Gao et al., 2021) because of its capability to model the interaction between the query and passage. Concretely, queries and passages are concatenated and fed into the BERT model, a pairwise score is then obtained by projecting the encoding of [CLS] token. The text ranking problem is cast as a binary classification problem. Nogueira et al. (2019) further proposed a pairwise BERT-based ranking model.

Recently, encoder-decoder language models, such as T5 (Raffel et al., 2020), have been adapted for text ranking. Nogueira et al. (2020) proposed a model that takes a query and passage pair as input of encoder, and the decoder produces the tokens "true" or "false" to indicate the relevance of a query and a given passage. Pradeep et al. (2021) further proposed a pairwise ranking model that takes a query and two passages as input and the decoder produces the token "true" if the first passage is more relevant then the second passage, and "false" otherwise. Zhuang et al. (2023b) proposed T5 encoder-only and encoder-decoder rerankers that optimize ranking performance directly by outputting real-value scores and using ranking losses. Despite the above-mentioned models which fine tune the pre-trained language model, some work proposed to use pre-trained language model directly. For example, Muennighoff (2022) proposed SGPT that uses GPT as reranking model directly; and Sachan et al. (2022) proposed UPR that uses a zero-shot question generation model via prompting a large language model in order to directly rerank passages.

We focus on the fine-tuning models in this work. As shown above, most progress has been made on the model structures. There are limited studies on training strategies. Most existing work use either annotated training data (Nogueira and Cho, 2019; Nogueira et al., 2020; Pradeep et al., 2021) or candidates generated by the first-stage retriever (Huang et al., 2020; Ren et al., 2021; Bonifacio et al., 2022; Zhuang et al., 2023b). Gao et al. (2021) quantitatively studies the benefits of sampling negative training examples from the first-stage retriever and in addition proposed a contrastive form loss.

## 3. Reranking Model

Given a query $q_i$ and a list of candidate passages $C(i) = c_1, c_2, ..., c_n$ in a document collection $D$, the ranking task aims to sort passages in the $C(i)$ such that more relevant passages have higher scores. More formally, we aim to learn a scoring function $s(q_i, c_j)$ such that $c^* = argmax_{j \in C(i)} s(q_i, c_j)$ is the most relevant passage to the query.

**Model structure** We follow Zhuang et al. (2023b) to use a T5-based cross-attention model. Specifically, we represent the query-passage pair as input sequence "Query: {Query} Document: {Title. Passage}" and feed it into the encoder. The output of the encoder is the encodings of the input sequence. We then apply a projection layer on the encoding of the first token and the output is used as the score. We use the encoder and discard the decoder allowing us to exploit the encoder-decoder pretraining while not requiring a decoder for inference. During inference, we pair query $q_i$ with each passage in $C(i)$ and compute scores. The ranking result is obtained by sorting the passages based on their scores.

The loss function we use is a listwise softmax cross entropy loss (Bruch et al., 2019) and is defined as follows:

$$\ell = -\sum_{i=1}^{n} \hat{y}_{ij} \log \left( \frac{e^{s_{ij}}}{\sum_{j'} e^{s_{ij'}}} \right) \quad (1)$$

where $s_{ij}$ is the predicted ranking score on query $q_i$ and passage $c_j$, and $\hat{y}_{ij}$ is the relevance label.

**Hybrid infused training data generation** During training, the construction of $C(i)$ is critical and affects the performance of the ranking model. $C(i)$ typically contains one relevant passage and a few non-relevant passages. The relevant passage is given, and the commonly used strategy is to sample non-relevant passages returned by a retriever.

In the pipelined multi-stage retrieval system consisting of retrieval and reranking stages, the list of non-relevant passages is usually formed by using the first-stage retriever (Bonifacio et al., 2022; Zhuang et al., 2023b). However, a better first-stage retriever does not always ensure a better training set for reranking models. In this work, we present a strategy to select better examples to train robust rerankers.

We use a hybrid retriever to generate passage lists. Specifically, we use a BM25 (Robertson et al., 1994) model as the sparse retrieval model and a T5-based dual encoder model (Ni et al., 2022) as the dense retrieval model. For each passage (and query), we concatenate the encodings from the two models to create a hybrid encoding. We perform maximal inner-product search (MIPS) using approximate nearest neighbor search over these hybrid encodings. This results in a different set of neighbors than independently selecting neighbors from BM25 and the T5 dual encoder.

To generate the training data for reranking model, we apply the hybrid retriever to the queries in the training set and retrieve top-K passages. We then sample $m$ negatives from retrieved result. In result, $C(i)$ is a passage list of size $m+1$ with one positive and $m$ negatives. $K$ and $m$ are hyperparameters and can be tuned based on each task.

Note that the choice of the sparse retriever and dense retriever for building the hybrid retriever is not limited to BM25 and dual encoder. It can be any sparse retriever and dense retriever as long as we can represent query and document as real-valued vectors.

## 4. Experimental Setup

We evaluate our proposed approach on two settings: one is supervised retrieval using MS MARCO where labeled relevant passages of a given query are available; the other is the zero-shot retrieval on BEIR where no labeled data is available in the target domains.

**MS MARCO passage ranking** This task aims to retrieve passages from a collection of web documents containing about 8.8 million passages. All questions in this dataset are sampled from real and anonymized Bing queries (Nguyen et al., 2016). The dataset contains 532,761 and 6980 examples in the training and development set respectively. Each query has one annotated relevant passage in average. We use them as positive training examples. We report our results using MRR@10 metric on the development set. The BM25 model in our hybrid retriever is a unigram model. We use the WordPiece tokenizer and vocabulary from uncased $\text{BERT}_{\text{base}}$ of size 30522. We use K=0.9 and b=0.8. We use GTR-Large model from Ni et al. (2022) as the dual encoder used in the hybrid retriever.

**Zero-shot retrieval** We also perform evaluation on the BEIR corpus (Thakur et al., 2021), a benchmark for zero-shot evaluation, to understand how our approach generalizes to out-of-domain setting. BEIR contains 18 evaluation datasets across 9 domains and no training data is available for those datasets.

The BM25 model used in this evaluation is the same as the one used for MS MARCO evaluation. Since we do not have training data, to train the dual encoder used for hybrid retriever, we follow Ma et al. (2021) to generate synthetic training data from a query generator and extend with iterative training following (Dai et al., 2023). Specifically, we pretrain the T5 based dual encoder on C4 dataset (Raffel et al., 2020) with the independent cropping task (Izacard et al., 2022). We then fine-tune the dual encoder using synthetically generated queries. The dual encoder structure is the same as GTR-Large model used for MS MARCO evaluation.

We apply a query generation model on the passages in the target corpus to generate (synthetic query, passage) pairs. The model is created by fine-tuning a general T5 model using question and passage pairs from Natural Question (NQ) (Kwiatkowski et al., 2019). Similar to PAQ (2021), we perform targeted generation, where knowing the location of the answer in a passage is important. Particularly, we form the input of the encoder as "Generate question >>> {title}.{passage} >>> {target sentence}", and the output of the decoder is the corresponding question. Here "target sentence" is the sentence that contains the short answer span, and "passage" corresponds to long answer and the passage of NQ. At inference time, for each dataset, we iterate over every passage and treat every sentence as the target sentence to generate synthetic queries. For large datasets, such as BioASQ and Climate-fever, we randomly sample 2 million passages for query generation.

Results are obtained using the official TREC evaluation tool[1]. We report normalised cumulative discount gain (nDCG@10) for all datasets.

**Implementation** The reranking models were initialized from T5 Version 1.1 models, and we evaluated on two sizes namely $\text{T5}_{\text{Base}}$ 1.1 and $\text{T5}_{\text{Large}}$ 1.1. Since we only use the encoders, the number of parameters are approximately 125M for Base model and 400M for Large model. We sampled 50 negative examples from top 250 retrieved passage for MS MARCO and from top retrieved passages from rank 10 to rank 210 for BEIR. The top 10 retrieved passages are filtered due to the possibility of false positives. We use input sequence length as 512 for all datasets except ArguAna, for which we use 1024. We train the models for 20000 steps

[1] https://github.com/usnistgov/trec_eval

| | Model size | MRR@10 |
|---|---|---|
| BM25 Anserini | | 0.1874 |
| HLATR | $\text{RoBERTa}_{\text{Large}}$ | 0.3680 |
| MiniLM | Distilled BERT | 0.3901 |
| monoT5 | $\text{T5}_{\text{3B}}$ | 0.3980 |
| RankT5-EncDec | $\text{T5}_{\text{Large}}$ | 0.3986 |
| $\text{DERR}_{\text{MS}}$ | $\text{T5}_{\text{Large}}$ 1.1 | 0.4222 |
| $\text{HYRR}_{\text{MS}}$ | $\text{T5}_{\text{Large}}$ 1.1 | **0.4235** |

Table 1: Reranking performance on MS MARCO Dev set in MRR@10.

| | Retriever | Reranker | | | |
|---|---|---|---|---|---|
| | BM25 | MiniLM | $\text{HYRR}_{\text{MS}}$ | HYRR | HYRR |
| **Model Size** | Anserini | 22M | 400M | 125M | 400M |
| NQ | 0.329 | 0.533 | **0.569** | 0.532 | 0.555 |
| MS MARCO | 0.228 | 0.413‡ | **0.435**‡ | 0.307 | 0.309 |
| Trec-Covid | 0.656 | 0.757 | 0.798 | 0.796 | **0.820** |
| BioASQ | 0.465 | 0.523 | **0.554** | 0.551 | 0.549 |
| NFCorpus | 0.325 | 0.350 | 0.371 | 0.379 | **0.382** |
| HotpotQA | 0.603 | 0.707 | **0.717** | 0.706 | 0.707 |
| FiQA-2018 | 0.236 | 0.347 | 0.411 | 0.408 | **0.437** |
| Signal-1M | 0.330 | **0.338** | 0.264 | 0.307 | 0.318 |
| Trec-News | 0.398 | 0.431 | 0.452 | 0.437 | **0.453** |
| Robust04 | 0.407 | 0.475 | 0.505 | 0.501 | **0.544** |
| ArguAna | **0.414** | 0.311 | 0.351 | 0.344 | 0.342 |
| Touché-2020 | 0.367 | 0.271 | **0.467** | 0.368 | 0.384 |
| Quora | 0.789 | 0.825 | 0.637 | 0.861 | **0.867** |
| DBPedia-entity | 0.313 | **0.409** | 0.402 | 0.385 | 0.403 |
| SCIDOCS | 0.158 | 0.166 | 0.184 | 0.183 | **0.187** |
| Fever | 0.753 | 0.819 | 0.825 | **0.868** | 0.861 |
| Climate-Fever | 0.213 | 0.253 | 0.262 | 0.272 | **0.294** |
| SciFact | 0.665 | 0.688 | 0.745 | 0.734 | **0.754** |
| CQADupStack | 0.299 | 0.370 | 0.368 | 0.398 | **0.416** |
| Average | 0.418 | 0.473 | 0.490 | 0.491 | 0.504 |
| Average w/o NQ | 0.423 | 0.470 | 0.486 | 0.489 | 0.501 |
| Avg. improvement on BM25 | | 4.63% | 6.26% | 6.58% | 7.81% |

Table 2: Reranking performance on BEIR in NDCG@10. ‡ indicates the in-domain performances. The results of baseline models are copied verbatim from the original papers. All models rerank the top-100 passages from BM25.

with batch size 64.

We implement the models using T5X[2] and we also use RAX (Jagerman et al., 2022), a learning-to-rank framework for implementing the ranking losses in reranking models. For training, it takes about 6 hours to train a dual encoder model and 6.5 hours to train a reranking model of T5-Large size using Cloud TPU-V3.

## 5. Results and Discussion

**MS MARCO results** To understand the effectiveness of our proposed approach, we fix the first-stage retrieval system and compare the reranking performance. Table 1 shows the performance of our proposed reranker on reranking BM25 top-1000 results. The BM25 results in row 1 is obtained from Anserini (Yang et al., 2017) toolkit[3] with parameters: k=0.82, b=0.68 following other baselines. The results of several strong baselines are shown in row 2-6. **HLATR** (Zhang et al., 2022) extends the retrieval-and-rerank pipeline with an additional ranking module by using the features from retrieval and reranking stages. It achieves top performance on MS MARCO leaderboard. **MiniLM** (Wang et al., 2020), which is a cross-encoder reranking model distilled from an ensemble of three teacher models. The other baselines are T5-based models: **monoT5** (Raffel et al., 2020; Rosa et al., 2022) and **RankT5-EncDec** (Zhuang et al., 2023b) adopt the encoder-decoder architecture. Our model adopts RankT5's encoder-only variant as described in Section 3. To compare with RankT5 model fairly, we implement our version: $\textbf{DERR}_{\textbf{MS}}$, which shares the same architecture and parameter settings as $\textbf{HYRR}_{\textbf{MS}}$. They only differ from the training data generation. Our reproduced $\text{DERR}_{\text{MS}}$ generates training data from dual encoder retriever. From row 7, we can see that $\text{HYRR}_{\text{MS}}$ outperforms all baselines. It is worth noting that the quality of the training set for the reranking model is critical. As can be seen, although monoT5 uses a much larger and more powerful $\text{T5}_{\text{3B}}$ model, it uses less carefully selected annotated negatives, and it performs worse than our reranker, which is much smaller in size. This demonstrates that our proposed training framework is effective in the supervised setting.

**BEIR results** Similar as evaluation on MS MARCO, we fix the first-stage retrieval system and compare the reranking performance. Table 2 shows the reranking performance of our proposed reranker. The BM25 results in Col.1 are obtained from Anserini toolkit with parameters: k=0.9 and b=0.4 following other baselines.

Col.2 and 3 show two supervised reranking models, which are trained on MS MARCO and perform inference on the target domains directly. $\textbf{HYRR}_{\textbf{MS}}$ is the model trained using our proposed approach on MS MARCO from Table 1. The results on MS MARCO are considered as in-domain for these models. We also show results of two models trained with synthetic data using our proposed method in Col.4 and 5., one is of size $\text{T5}_{\text{Base}}$ 1.1 and one is of size $\text{T5}_{\text{Large}}$ 1.1. The results demonstrate the effectiveness of our proposed method in zero-shot settings. We note that the results on NQ cannot be considered completely out-of-domain since the question generation model used to generate training data for the hybrid retriever is trained on the NQ dataset.

**Ablation** To show the robustness of HYRR, we conduct an ablation experiment. We train rerankers using the training data generated from the BM25 or the dual encoder model, namely **BM25RR** and **DERR**. Those two variants are commonly seen in many pipelined retrieval systems, where rerankers are simply trained upon the first-stage retriever. We

[2]https://github.com/google-research/t5x

[3]https://github.com/castorini/anserini

| Retriever ↓ | No Reranker | BM25RR | DERR | HYRR |
|---|---|---|---|---|
| | | **MS MARCO** | | |
| BM25 | 0.187 | 0.375 | 0.422 | **0.424** |
| DE | 0.378 | 0.350 | **0.440** | **0.440** |
| Hybrid | 0.390 | 0.351 | 0.438 | **0.440** |
| | | **SciFact** | | |
| BM25 | 0.677 | 0.750 | 0.742 | **0.752** |
| DE | 0.597 | **0.755** | 0.745 | 0.752 |
| Hybrid | 0.706 | 0.753 | 0.744 | **0.759** |

Table 3: Ablation results on MS MARCO in MRR@10 and SciFact in nDCG@10.

train them using the same training setting for HYRR and then apply them on three retrievers: the BM25 model, the dual encoder model (DE) and the hybrid retriever, respectively. We experiment on both supervised setting and zero-shot setting. The results on MS MARCO are shown in the top section of Table 3. As we can see HYRR provides the most performance gain over all three retrievers on MRR@10. The BM25RR improves the performance on BM25 while hurts the other two. DERR achieves best performance when we apply it to DE. It also improves the other two retrievers but not as much as HYRR. This shows that HYRR not only outperforms the other two rerankers but also is effective on different retrievers. We pick SciFact from BEIR as an example for zero-shot setting. The results are shown in bottom part of Table 3, and we observe the similar trends on other datasets in BEIR. Similarly, HYRR improves both nDCG@10 over all three retrievers. This believe is the evidence that the robustness of the training data for the reranker is carried over to the robustness of the reranker itself.

In addition, to understand the benefit to use hybrid retriever for training data generation, we conduct another ablation experiment. We mix the training data generated from the BM25 and the dual encoder model in 1:1 ratio and train a reranker. We evaluate on MS MARCO and the model achieves 0.417 in MRR@10 when reranking BM25 top 1000. When comparing with the results in row 1 from Table 3, we can see that our approach significantly outperforms the approach that simply applying training data ensemble.

## 6. Conclusion

We proposed a generic training framework for rerankers in the two-stage retrieval pipeline. The reranker is a neural cross-attention model which learns from negatives examples generated by a hybrid retriever, which is composed of term-based and neural retrievers. The proposed approach is robust and outperforms several strong baselines on MS MARCO and BEIR benchmark dataset, demonstrating its practicality and generalizability.

## 7. Acknowledgements

We thank the three anonymous reviewers for their insightful comments, Don Metzler for reviewing the manuscript.

## 8. Bibliographical References

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Data augmentation for information retrieval using large language models. *arXiv preprint arXiv:2202.05144*.

Sebastian Bruch, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2019. An analysis of the softmax cross entropy loss for learning-to-rank with binary relevance. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '19, page 75–78. Association for Computing Machinery.

Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith Hall, and Ming-Wei Chang. 2023. Promptagator: Few-shot dense retrieval from 8 examples. In *The Eleventh 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. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. 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.

Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2553–2561.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand

Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Rolf Jagerman, Xuanhui Wang, Honglei Zhuang, Zhen Qin, Michael Bendersky, and Marc Najork. 2022. Rax: Composable learning-to-rank using jax. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 3051–3060. Association for Computing Machinery.

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.

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.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Alexander Cosgrove, Christopher D Manning, Christopher Re, Diana Acosta-Navas, Drew Arad Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue WANG, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Andrew Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research*.

Ji Ma, Ivan Korotkov, Yinfei Yang, Keith B. Hall, and Ryan T. McDonald. 2021. Zero-shot neural passage retrieval via domain-targeted synthetic question generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1075–1088.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document reranking with a large language model. *arXiv preprint arXiv:2305.02156*.

Niklas Muennighoff. 2022. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*.

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 *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. 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 (EMNLP)*.

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

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

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*.

Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *arXiv preprint arXiv:2101.05667*.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, et al. 2023. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563*.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. 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.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. 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.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pages 109–126. National Institute of Standards and Technology (NIST).

Guilherme Moraes Rosa, Luiz Bonifacio, Vitor Jeronymo, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. 2022. No parameter left behind: How distillation and model size affect zero-shot retrieval. *arXiv preprint arXiv:2206.02873*.

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.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937.

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)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008. Curran Associates, Inc.

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. volume 33, pages 5776–5788.

Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pages 1253–1256.

Yanzhao Zhang, Dingkun Long, Guangwei Xu, and Pengjun Xie. 2022. Hlatr: Enhance multi-stage text retrieval with hybrid list aware transformer reranking.

Honglei Zhuang, Zhen Qin, Kai Hui, Junru Wu, Le Yan, Xuanhui Wang, and Michael Berdersky. 2023a. Beyond yes and no: Improving zero-shot llm rankers via scoring fine-grained relevance labels. *arXiv preprint arXiv:2310.14122*.

Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2023b. 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. Association for Computing Machinery.