



# Aligning Web Query Generation with Ranking Objectives via Direct Preference Optimization

João Coelho  
Carnegie Mellon University  
Pittsburgh, Pennsylvania, United States  
jmcoelho@andrew.cmu.edu

João Magalhães  
NOVA School of Science and Technology  
Lisbon, Portugal  
jmag@fct.unl.pt

Bruno Martins  
Instituto Superior Técnico and INESC-ID  
Lisbon, Portugal  
bruno.g.martins@tecnico.ulisboa.pt

Chenyan Xiong  
Carnegie Mellon University  
Pittsburgh, Pennsylvania, United States  
cx@cs.cmu.edu

## Abstract

Neural retrieval models excel in Web search, but their training requires substantial amounts of labeled query-document pairs, which are costly to obtain. With the widespread availability of Web document collections like ClueWeb22, synthetic queries generated by large language models offer a scalable alternative. Still, synthetic training queries often vary in quality, which leads to suboptimal downstream retrieval performance. Existing methods typically filter out noisy query-document pairs based on signals from an external re-ranker. In contrast, we propose a framework that leverages Direct Preference Optimization (DPO) to integrate ranking signals into the query generation process, aiming to directly optimize the model towards generating high-quality queries that maximize downstream retrieval effectiveness. Experiments show higher ranker-assessed relevance between query-document pairs after DPO, leading to stronger downstream performance on the MS MARCO benchmark when compared to baseline models trained with synthetic data.

## CCS Concepts

• Information systems → Retrieval models and ranking.

## Keywords

Dense Retrieval, Query Generation, Synthetic Data

### ACM Reference Format:

João Coelho, Bruno Martins, João Magalhães, and Chenyan Xiong. 2025. Aligning Web Query Generation with Ranking Objectives via Direct Preference Optimization. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25)*, July 13–18, 2025, Padua, Italy. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3726302.3730162>

## 1 Introduction

The effective training of neural retrieval models benefits from large-scale query-document relevance annotations [8]. However, acquiring such data remains a bottleneck due to its high costs.

To address this limitation, recent work has explored using Large Language Models (LLMs) to generate synthetic queries from documents [6, 18, 35], enabling retrieval models to be trained without full reliance on human annotations. Combined with the availability of large Web corpora such as ClueWeb22 [27], this establishes LLM query generators as a scalable approach for training data creation. Despite this potential, the quality of synthetic queries varies significantly, with studies showing that it still lags behind human-annotated data [8], which degrades downstream effectiveness [10].

To mitigate this problem, approaches for query generation consider post hoc filtering, e.g., using an external ranking signal [15] or through consistency checking during negative sampling [6]. While this improves query quality, it does not guide the query generator toward producing queries that maximize retrieval performance.

In this work, instead of post hoc query filtering, we propose a framework that directly integrates ranking feedback into the query generation process using Direct Preference Optimization (DPO) [28]. To this end, we generate multiple queries per document, score them using a ranking model, and build a preference dataset to optimize the query generator toward producing high-quality queries. This approach enables the model to learn directly from ranking signals, aiming to improve the overall quality of queries used for dense retrieval training. More specifically, we independently consider ranking rewards from point-wise re-rankers and list-wise LLM prompting techniques, demonstrating the framework's flexibility and opening future directions for reward selection.

We conduct experiments on the MS MARCO dataset [23], i.e. a standard benchmark for passage and document retrieval, to evaluate the impact of DPO-optimized queries on retrieval performance. Our framework demonstrates three key contributions: (1) a training paradigm that directly aligns query generation with ranking objectives, diminishing the need for post hoc filtering inefficiencies due to increased ranker-assessed relevance; (2) flexibility in integrating diverse ranking signals, including both point-wise re-rankers and LLM-based list-wise prompting; and (3) empirical validation, showing that retrievers trained with synthetic data from our generation pipeline outperform other open-source alternatives in the Web domain. Our models and code are available in a public GitHub repository: <https://github.com/cxcscmu/dpo-query-generation>.



This work is licensed under a Creative Commons Attribution 4.0 International License. *SIGIR '25, Padua, Italy*

© 2025 Copyright held by the owner/author(s).  
ACM ISBN 979-8-4007-1592-1/2025/07  
<https://doi.org/10.1145/3726302.3730162>

## 2 Related Work

Transformer-based bi-encoders are the standard architecture for dense retrieval [16, 38], typically trained with contrastive objectives and hard negative mining strategies such as ANCE [38]. Retrieval-aligned pre-training on in-domain corpora is also commonly adopted to improve retrieval effectiveness [9, 19–22, 36].

The community has explored generating synthetic queries from documents for model training. Approaches such as Doc2Query [25] and DocT5Query [24] train a lightweight Transformer [33] on labeled query–document pairs to expand document representations. Subsequent work [10] demonstrates that filtering out hallucinated queries can further enhance downstream performance. More recent methods leverage LLMs. For instance, InPars [2] employs few-shot prompting with filtering based on generation probability, while InPars-v2 [15] leverages a supervised ranker for query filtering. Similarly, Promptagator [6] uses task-specific prompts to perform few-shot query generation. Building on these approaches, the Gecko model [18] iteratively refines synthetic queries through a process involving retrieval, re-ranking, positive relabeling, and hard negative sampling, thereby improving the quality of the training data.

Recent work has explored reinforcement learning to improve synthetic query generation. Token-level Proximal Policy Optimization (TPPO) [26, 30] has been applied to query suggestion tasks, optimizing generation based on token-level rewards derived from user interaction histories. While this setting targets interactive scenarios rather than offline generation from documents, it highlights the promise of reinforcement learning for enhancing query quality.

## 3 Methodology

This section describes our query generation process, then details the training of retrieval models using the resulting synthetic queries.

### 3.1 Query Generation

We start from a baseline query generator  $\mathcal{M}_g$  (e.g., an LLM of choice or an existing model fine-tuned for query generation). To synthesize queries, we follow a contrastive prompting strategy [37] where given a document  $d$ , a negative example  $d^-$  is sampled from a corpus. Then, the model is prompted to generate a query  $q$  that is specific to  $d$ , while being less relevant to  $d^-$ , i.e.  $q \sim \mathcal{M}_g(d, d^-)$ . For simplicity, we represent this generation process as  $\mathcal{M}_g(d)$ , assuming implicit contrastive prompting.

By aiming to increase  $(q, d)$  similarity, this strategy mitigates retrieval training scenarios where poorly generated queries allow other corpus documents to dominate over the original positive during negative sampling, which is a source of noise.

### 3.2 Aligning the Query Generator

Instead of filtering queries generated by  $\mathcal{M}_g$ , we hypothesize that the query generator can be guided towards better downstream retrieval performance. To this end, we propose leveraging a reward model to obtain a preference dataset for DPO alignment. Given the baseline generator  $\mathcal{M}_g$  and a corpus  $\mathcal{C}$ , we start by sampling a random pool of documents from  $\mathcal{C}$ , denoted as  $\mathcal{D}$ . For each document in  $\mathcal{D}$ , we use  $\mathcal{M}_g$  to generate a set of  $n$  synthetic queries:

$$Q_d = \{q_{d,1}, \dots, q_{d,n}\} \sim \mathcal{M}_g(d), \forall d \in \mathcal{D}. \quad (1)$$

**Obtaining Rewards:** As a first reward mechanism, we consider the usage of a ranking model  $\mathcal{R}$  to score each individual pair, since previous work has established this as a strong filtering method [15]. More specifically, we used `bge-reranker-v2-m3`<sup>1</sup>, i.e. a small cross-encoder model available on HuggingFace, which offers a good trade-off between retrieval performance and computation resources.

By applying  $\mathcal{R}$  on the synthetic pairs, we obtain triplets in the form  $\{(q, d, \mathcal{R}(q, d)) \mid d \in \mathcal{D}, q \in Q_d\}$ . We can then extract a preference dataset  $\mathcal{D}_{rr}$ , where, for each document, a random pair of queries is sampled such that one has a higher score than the other:

$$\mathcal{D}_{rr} = \{(d, q^+, q^-) \mid d \in \mathcal{D}, (q^+, q^-) \in Q_d, \mathcal{R}(q^+, d) > \mathcal{R}(q^-, d)\}. \quad (2)$$

Alternatively, we can consider list-wise reward models, for instance through GPT3.5 prompting, which previous work has established as a competitive ranking method [31]. In this setup, given  $d$  and  $Q_d$ , the model  $\mathcal{G}$  is prompted to explicitly select the best and worst queries from  $Q_d$ , producing a preference dataset  $\mathcal{D}_{gpt}$ :

$$\mathcal{D}_{gpt} = \{(d, q^+, q^-) \mid d \in \mathcal{D}, (q^+, q^-) \sim \mathcal{G}(d, Q_d)\}. \quad (3)$$

**Fine-tuning the Generator:** Both  $\mathcal{D}_{rr}$  and  $\mathcal{D}_{gpt}$  allow training  $\mathcal{M}_g$  following the Direct Preference Optimization (DPO) loss [28]:

$$\mathcal{L} = -\mathbb{E}_{(d, q^+, q^-) \sim \mathcal{D}_{pref}} \left[ \log \sigma \left( \beta \log \frac{\mathcal{M}_g^*(q^+ \mid d)}{\mathcal{M}_g(q^+ \mid d)} - \beta \log \frac{\mathcal{M}_g^*(q^- \mid d)}{\mathcal{M}_g(q^- \mid d)} \right) \right], \quad (4)$$

where  $\mathcal{D}_{pref}$  is the preference dataset (e.g.,  $\mathcal{D}_{rr}$  or  $\mathcal{D}_{gpt}$ ), and  $\beta$  is an hyper-parameter for tuning. This aims to ensure that the likelihood of generating ranker-preferred queries is higher, thereby directly including the ranking signals into the generation process rather than as a filtering stage, which we argue contributes to more robust large-scale generation.

### 3.3 Dense Retriever Training

In order to obtain positive pairs for dense retriever training, we can use the aligned query generator  $\mathcal{M}_g^*$ , to conduct inference on a target Web document corpus. Given a random sample of  $k$  documents,  $\mathcal{D}_k$ , we use  $\mathcal{M}_g^*$  to sample one query per document, obtaining a set of synthetic positive pairs:

$$\mathcal{D}_{train} = \{(q, d) \mid d \in \mathcal{D}_k, q \sim \mathcal{M}_g^*(d)\}. \quad (5)$$

For the embedding model ( $\mathcal{M}_e$ ), we leverage Qwen2.5-0.5B [39] as the language model backbone. Following previous work on decoder-only dense retrievers, we apply a bi-directional attention mask and conduct masked next token prediction as continual pre-training. This has been shown to better align the model to the task of generating dense representations, with continual pre-training being necessary for the model to adapt to the new mask [1]. We leverage 100k random documents from MS MARCO [23] for pre-training, as a form of domain adaptation [9].

Given  $\mathcal{D}_{train}$ , we follow a standard setup, where an external model is used to sample hard negatives for each query. More specifically, given the top-100 documents for a query, 5 negatives are sampled from the documents that are ranked below the original positive [17]. We also consider the relabeling mechanism proposed by Gecko [18], where if the original positive document is not in the top-100, the top-1 is considered as the new positive document instead of discarding the query.

<sup>1</sup><https://huggingface.co/BAAL/bge-reranker-v2-m3>

**Table 1: Performance of multiple models trained with synthetic data on Web benchmarks, including model size and number of training pairs. \*Synthetic-only checkpoint unavailable for reproduction; MARCO passage result extracted from original paper.**

Model	Size Pairs		Datasets			
			MARCO Doc	MARCO Passage	TREC-DL 19 Passage	TREC-DL 20 Passage
			MRR@100	nDCG@10	nDCG@10	nDCG@10
BM25 [29]	-	-	0.2300	0.2280	0.5058	0.4796
Contriever [13]	0.1B	700M	0.2322	0.2060	0.4794	0.4352
WebDRO [11]	0.2B	13.8M	0.3030	0.2741	0.4981	0.4591
E5 <sub>large</sub> [34]	0.3B	270M	0.3039	0.2620	0.5108	0.5024
MiniCPM-Embedding <sup>2</sup>	2.4B	2M	0.3180	0.2980	0.5730	<b>0.5646</b>
E5 <sub>Mistral</sub> [35]*	7B	0.5M	-	0.2570	-	-
Ranker-DPO	0.5B	1M	<u>0.3727</u>	<u>0.3286</u>	<u>0.6171</u>	0.5337
GPT-DPO	0.5B	1M	<b>0.3795</b>	<b>0.3340</b>	<b>0.6213</b>	<u>0.5496</u>

We fine-tune  $\mathcal{M}_e$  leveraging in-batch and cross-GPU negatives, aiming to minimize the InfoNCE loss [32]:

$$\mathcal{L} = -\log \frac{e^{\cos(\mathcal{M}_e(q), \mathcal{M}_e(d))}}{e^{\cos(\mathcal{M}_e(q), \mathcal{M}_e(d))} + \sum_{n \in \mathcal{N}} e^{\cos(\mathcal{M}_e(q), \mathcal{M}_e(n))}}, \quad (6)$$

where  $\mathcal{N}$  is the set of all negatives. Representations  $\mathcal{M}_e(\cdot)$  are obtained by average pooling and compared using the cosine similarity.

## 4 Experiments

This section describes our experimental setup, evaluation, and results. We begin by detailing the configurations for both query generation and retrieval training, along with an overview of the evaluation benchmarks. Next, we present the experimental results, assessing the impact of our query generation strategy and conducting scaling experiments in terms of the number of documents.

### 4.1 Setup and Implementation Details

For our corpus  $C$ , we consider ClueWeb22-B [27], which contains the most popular and frequently visited web pages of ClueWeb22, making it well-suited for query generation due to its inherent informativeness. We leverage a sample of 35 million documents to ensure broad coverage while maintaining computational efficiency.

For query generation, we leverage an open-source model as our baseline  $\mathcal{M}_g$ , which was used to obtain the synthetic data to train the MiniCPM-Embedding model<sup>2</sup>. This generator was obtained by fine-tuning LLaMA-3-7B [7] with synthetic query-document pairs obtained from GPT-4, following the contrastive prompting strategy described in Section 3.1.

Starting with a pool of 100k random documents from  $C$ , we sample 5 queries per document using  $\mathcal{M}_g$ , and use this data to follow the process described in Section 3.2, in order to obtain the improved generator  $\mathcal{M}_g^*$ . The document count (100k) was selected as a practical upper bound that allows model convergence within reasonable training cost. Due to the size of the model, DPO training is conducted using LoRA [12], considering 8 NVIDIA L40S GPUs, a batch size of 8, and a learning rate of 1e-5. This setup induces a one-time 10-hour overhead compared to just using  $\mathcal{M}_g$ .

<sup>2</sup><https://huggingface.co/openbmb/MiniCPM-Embedding>

After obtaining the aligned query generator, we randomly sample documents from  $C$  (excluding the previous 100k documents used for model training) and generate one query per document. The obtained query-document positive pairs are then used to train two retrieval models following the setup introduced in Section 3.3, independently leveraging synthetic data obtained from aligning the generator with ranker or GPT rewards. We adopt the same embedding-based negative sampling strategy used in MiniCPM-Embedding, by using an external model to obtain negatives from  $C$ . Training is conducted for one epoch with a batch size of 64, a learning rate of 1e-4, and distributed across 8 L40S NVIDIA GPUs.

### 4.2 Evaluation

For evaluation, we leverage MS MARCO [23], i.e. a standard Web benchmark including both document and passage retrieval, as well as the queries from the TREC-DL 2019 and 2020 passage tasks. Regarding metrics, we follow common practice: for MS MARCO document retrieval we use the Mean Reciprocal Rank (MRR) [3]; for MS-MARCO passage retrieval and associated TREC-DL tracks [4, 5], we use the normalized Discounted Cumulative Gain (nDCG) [14].

### 4.3 Retrieval Performance

Table 1 presents a comparative evaluation of our models against open-source approaches leveraging synthetic data. Specifically, we benchmark our models against several baselines: lexical retrieval (BM25), models trained on anchor data (WebDRO), models trained on heuristically selected pairs from structured documents (E5<sub>large</sub>), and models trained on LLM-generated data (E5<sub>Mistral</sub> and MiniCPM-Embedding). For MiniCPM-Embedding, we report results obtained using synthetic data exclusively, noting that this synthetic data was generated by the same generator employed as a baseline in this study. Overall, our methods (i.e., either using a ranker or GPT as reward models) yield superior performance on Web benchmarks, showcasing the effectiveness of the proposed framework. Both reward mechanisms demonstrate similar performance, likely due to the fact that both rely on semantic cues when distinguishing between queries. The next sections analyze query quality and document scale to elucidate the observed improvements.

**Table 2: Comparison between the baseline generator and the DPO variant, considering a pool of 250K generated queries and consistency filtering.**

	Initial Pool	Kept	MARCO Doc	
			MRR@100	R@1000
Before DPO	250K	156K (62%)	0.1635	0.5902
After DPO	250K	231K (92%)	0.2625	0.7491

**Table 3: Comparison of queries generated before and after DPO. Each pair was generated for the same document.**

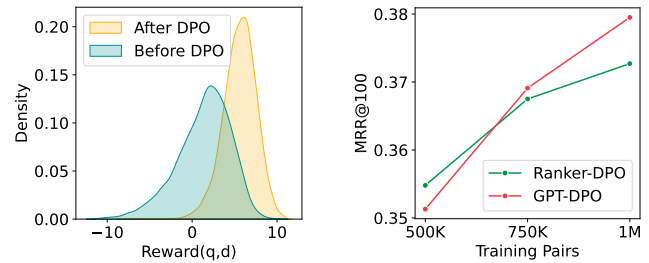
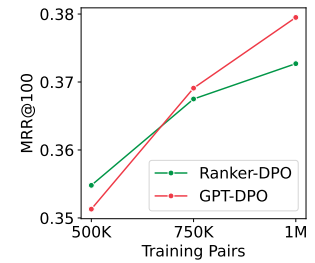
Before DPO	After DPO
What are the steps to adjust the idle on a 2003 Toyota Tacoma 2.4 liter?	Adjusting idle on 2003 Toyota Tacoma
What makes the 2019 Ram 1500's frame and hybrid system superior?	2019 Ram 1500 redesign and features
How do religious beliefs influence faith and practice during pandemic crises like COVID-19?	Impact of COVID-19 on religion and faith perception

#### 4.4 Query Retention

To evaluate the impact of generator alignment via DPO, we analyze synthetic query quality through query retention rate and ranker-assessed relevance. We focus on the ranker-DPO variant, as its alignment reward model  $\mathcal{R}$  provides relevance scores that enable comparisons between baseline and optimized queries. We generate 250k queries, before and after DPO, and train two dense retrieval models. Negative documents are sampled from the previously introduced corpus of 35 million candidates, making the generation task more demanding: suboptimal queries often fail to distinguish the original positive from high-relevance negatives introduced by large-scale sampling, which can contribute to training noise.

Table 2 shows that after DPO, following a standard consistency parsing mechanism that discards queries if the original positive document is not in the top-100 when sampling negatives, queries have an approximately 50% higher retention rate. This high difference indicates that integrating ranking feedback via DPO guides the query generation process toward higher relevance, hence improving both data efficiency and quality, given the increase in retrieval performance and query retention.

To further consolidate these findings, we examined the distribution of ranker reward scores  $\mathcal{R}(q, d)$  for a sample of 10k query-document pairs. As depicted in the KDE plot in Figure 1, the post-DPO score distribution exhibits a pronounced rightward shift relative to the one from the base generator. This reflects an improvement in query-document relevance, thereby validating the efficacy of our DPO framework in steering the generator toward higher-quality outputs. As illustrated in Table 3, the DPO-aligned generator produces more concise and web-like queries, transforming verbose questions into focused search terms while preserving the core information intent.

**Figure 1: Reward ( $\mathcal{R}$ ) distributions for 10k ( $q, d$ ) pairs, before and after DPO.****Figure 2: Performance on MARCO Doc as synthetic training pairs increase.**

#### 4.5 Scaling Synthetic Data

We also present an empirical investigation of how model performance scales with synthetic training pairs. Figure 2 demonstrates that both models exhibit progressive improvements in retrieval effectiveness as the document pools expand. While larger document pools consistently enhance model effectiveness, our results show that resource-constrained implementations remain viable, as competitive performance levels can be achieved under limited computational budgets, while maintaining the flexibility to scale.

### 5 Conclusions and Future Work

This paper introduces a framework that aimed at improving synthetic training data generation for dense retrieval, representing a shift from post hoc filtering methods. While our implementation focuses on Web retrieval, the principle of using preference optimization to align data generation with task performance can be applied to other systems that rely on synthetic data. Rather than treating query generation and filtering as separate steps, our results suggest that explicitly integrating ranking objectives into the generator training process leads to more robust and data-efficient systems. The empirical improvements we observe validate the proposed approach, pointing toward a promising direction for developing more effective synthetic data generators for dense retrieval training.

Future work may focus on rewards beyond the use of ranker and LLM-based ranking signals, e.g. by incorporating task-specific signals to promote generalization. In addition, while we currently employ a single model for query generation, using multiple models could enhance the diversity of synthetic data [7] and improve retrieval fine-tuning. Finally, considering the computational cost of query generation, developing smaller and more efficient generators without compromising quality is a worthwhile future direction.

### Acknowledgments

We thank the anonymous reviewers for their valuable comments and suggestions. This research was supported by the Portuguese Recovery and Resilience Plan through project C645008882-00000055 (i.e., the Center For Responsible AI), and also by the Fundação para a Ciência e Tecnologia, specifically through the project with reference UIDB/50021/2020, the project with reference UIDP/04516/2020, and through the Ph.D. scholarship with reference PRT/BD/153683/2021 under the Carnegie Mellon Portugal Program.

## References

- [1] Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. LLM2Vec: Large Language Models Are Secretly Powerful Text Encoders. *ArXiv abs/2404.05961* (2024).
- [2] Luiz Henrique Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Frassetto Nogueira. 2022. InPars: Data Augmentation for Information Retrieval using Large Language Models. *ArXiv abs/2202.05144* (2022).
- [3] Nick Craswell. 2009. Mean Reciprocal Rank. In *Encyclopedia of Database Systems*.
- [4] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 Deep Learning Track. In *Text REtrieval Conference (TREC)*.
- [5] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 Deep Learning Track. In *Text REtrieval Conference (TREC)*.
- [6] Zhuyun Dai, Vincent Y. Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B. Hall, and Ming-Wei Chang. 2023. Promptagator: Few-shot Dense Retrieval From 8 Examples. In *International Conference on Learning Representations (ICLR 2023)*.
- [7] Abhimanyu Dubey et al. 2024. The Llama 3 Herd of Models. *ArXiv abs/2407.21783* (2024).
- [8] Yan Fang, Jingtao Zhan, Qingyao Ai, Jiabin Mao, Weihang Su, Jia Chen, and Yiqun Liu. 2024. Scaling Laws For Dense Retrieval. In *International Conference on Research and Development in Information Retrieval (SIGIR 2024)*.
- [9] Luyu Gao and Jamie Callan. 2022. Unsupervised Corpus Aware Language Model Pre-training for Dense Passage Retrieval. In *Annual Meeting of the Association for Computational Linguistics (ACL 2022)*.
- [10] Mitko Gospodinov, Sean MacAvaney, and Craig Macdonald. 2023. Doc2Query-: When Less is More. In *European Conference on Information Retrieval (ECIR 2023)*.
- [11] Peixuan Han, Zhenghao Liu, Zhiyuan Liu, and Chenyan Xiong. 2024. Enhancing Dense Retrievers' Robustness with Group-level Reweighting. *ArXiv abs/2310.16605* (2024).
- [12] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations, (ICLR 2022)*.
- [13] 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* (2022).
- [14] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems* (2002).
- [15] Vitor Jeronymo, Luiz Henrique Bonifacio, Hugo Abonizio, Marzieh Fadaee, Roberto de Alencar Lotufo, Jakub Zavrel, and Rodrigo Frassetto Nogueira. 2023. InPars-v2: Large Language Models as Efficient Dataset Generators for Information Retrieval. *ArXiv abs/2301.01820* (2023).
- [16] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*.
- [17] Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models. *ArXiv abs/2405.17428* (2024).
- [18] Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernández Ábrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha Reddy Jonnalagadda, Ming-Wei Chang, and Iftikhar Naim. 2024. Gecko: Versatile Text Embeddings Distilled from Large Language Models. *ArXiv abs/2403.20327* (2024).
- [19] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *Annual Meeting of the Association for Computational Linguistics (ACL 2019)*.
- [20] Shuqi Lu, Di He, Chenyan Xiong, Guolin Ke, Waleed Malik, Zhicheng Dou, Paul Bennett, Tie-Yan Liu, and Arnold Overwijk. 2021. Less is More: Pretrain a Strong Siamese Encoder for Dense Text Retrieval Using a Weak Decode. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*.
- [21] Guangyuan Ma, Xing Wu, Zijia Lin, and Songlin Hu. 2024. Drop your Decoder: Pre-training with Bag-of-Word Prediction for Dense Passage Retrieval. *ArXiv abs/2401.11248* (2024).
- [22] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, and Xueqi Cheng. 2022. Pre-train a Discriminative Text Encoder for Dense Retrieval via Contrastive Span Prediction. In *International Conference on Research and Development in Information Retrieval (SIGIR 2022)*.
- [23] 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 *Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches*.
- [24] Rodrigo Nogueira and Jimmy Lin. 2019. *From doc2query to docTTTTTquery*. Technical Report. University of Waterloo.
- [25] Rodrigo Frassetto Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document Expansion by Query Prediction. *ArXiv abs/1904.08375* (2019).
- [26] Yichen Ouyang, Lu Wang, Fangkai Yang, Pu Zhao, Chenghua Huang, Jianfeng Liu, Bochen Pang, Yaming Yang, Yuefeng Zhan, Hao Sun, Qingwei Lin, Saravan Rajmohan, Weiwei Deng, Dongmei Zhang, Feng Sun, and Qi Zhang. 2024. Token-level Proximal Policy Optimization for Query Generation. *ArXiv abs/2411.00722* (2024).
- [27] Arnold Overwijk, Chenyan Xiong, and Jamie Callan. 2022. ClueWeb22: 10 Billion Web Documents with Rich Information. In *International Conference on Research and Development in Information Retrieval (SIGIR 2022)*.
- [28] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Conference on Neural Information Processing Systems (NeurIPS 2023)*.
- [29] Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* (2009).
- [30] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *ArXiv abs/1707.06347* (2017).
- [31] 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 *Conference on Empirical Methods in Natural Language Processing (EMNLP 2023)*.
- [32] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *ArXiv abs/1807.03748* (2018).
- [33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Conference on Neural Information Processing Systems (NeurIPS 2017)*.
- [34] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text Embeddings by Weakly-Supervised Contrastive Pre-training. *ArXiv abs/2212.03533* (2022).
- [35] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving Text Embeddings with Large Language Models. In *Annual Meeting of the Association for Computational Linguistics (ACL 2024)*.
- [36] Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. 2022. RetroMAE: Pre-Training Retrieval-oriented Language Models Via Masked Auto-Encoder. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2022)*.
- [37] Chenyan Xiong, Zhenghao Liu, Si Sun, Zhuyun Dai, Kaitao Zhang, Shi Yu, Zhiyuan Liu, Hoifung Poon, Jianfeng Gao, and Paul Bennett. 2020. CMT in TREC-COVID Round 2: Mitigating the Generalization Gaps from Web to Special Domain Search. *ArXiv abs/2011.01580* (2020).
- [38] 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 *International Conference on Learning Representations (ICLR 2021)*.
- [39] An Yang et al. 2024. Qwen2.5 Technical Report. *ArXiv abs/2412.15115* (2024).