Study on LLMs for Promptagator-Style Dense Retriever Training

Daniel Gwon*
Massachusetts Institute of
Technology Lincoln Laboratory
Lexington, MA, USA
daniel.gwon@ll.mit.edu

Nour Jedidi*†
University of Waterloo
Waterloo, ON, Canada
njedidi@uwaterloo.ca

Jimmy Lin
University of Waterloo
Waterloo, ON, Canada
jimmylin@uwaterloo.ca

Abstract

Promptagator [6] demonstrated that Large Language Models (LLMs) with few-shot prompts can be used as task-specific query generators for fine-tuning domain-specialized dense retrieval models. However, the original Promptagator approach relied on proprietary and large-scale LLMs which users may not have access to or may be prohibited from using with sensitive data. In this work, we study the impact of open-source LLMs at accessible scales (≤14B parameters) as an alternative. Our results demonstrate that open-source LLMs as small as 3B parameters can serve as effective Promptagator-style query generators. We hope our work will inform practitioners with reliable alternatives for synthetic data generation and give insights to maximize fine-tuning results for domain-specific applications.[1]

CCS Concepts

• **Information systems → Retrieval models and ranking**.

Keywords

Dense Retrieval, Document Representation, Contrastive Learning, Synthetic Data Generation, LLMs

1 Introduction

Dense retrieval models have greatly enhanced modern Information Retrieval (IR) systems [13], endowing search systems with semantic search capabilities while minimizing the latency associated with models using cross-encoder architectures [16]. However, for domains and tasks that lack large amounts of training data, dense retrievers have been shown to struggle, underperforming sparse bag-of-words techniques like BM25 [18].

To overcome this challenge, researchers have looked towards employing Large Language Models (LLMs) as dataset generators [2, 6]. These techniques typically prompt LLMs to generate synthetic queries that align with given documents from the corpus for training dense retrieval models. A popular approach is Promptagator [6], which leverages an LLM to amplify a few human-annotated examples from the target domain into many training examples. The core idea behind Promptagator is that different search tasks and domains have different relevance definitions that should be considered when generating synthetic training queries.

While Promptagator was shown to be highly effective in domain-specific settings, its results were based on a proprietary and large-scale LLM, which users may not have access to or may be prohibited from using with sensitive data. With advancements in open-source LLMs — particularly at accessible scales (≤14B parameters) — we investigate *Promptodile*, an open-source variant of Promptagator focusing on the utility of LLMs in three settings: (1) low-resource domains where training data doesn't exist, (2) domains where information needs are found in sensitive data (e.g., PII or proprietary data) that users can't share with popular LLM providers (e.g. OpenAI and Anthropic), and (3) compute-constrained environments where users may not have resources to self-host larger models.

Towards this goal, we study Promptodile with 10 open-source LLMs across four LLM families, ranging in scale from 1B-14B parameters, evaluated on seven low-resource BEIR datasets. Our findings can be summarized as follows: (1) Across all LLMs, Promptodile is generally competitive with Promptagator, demonstrating that recent, accessible LLMs can serve as effective query generators. We highlight that these results came *without* round-trip filtering, suggesting that as LLMs improve, filtering may be less important. (2) Furthermore, we find that smaller LLMs (≈3B parameters) are just as effective as larger ones (between 7B to 14B parameters), demonstrating that there is no benefit in leveraging expensive LLMs for query generation.

We emphasize that we are not the first to study open-source LLMs as query generators [3, 12]. Our research builds upon this line of work by focusing on dense retrievers and studying across a wider range of LLMs. The primary contribution of Promptodile is to provide insights for practitioners building Promptagator-style retrieval systems in low resource domains amidst compute and data privacy constraints; we do *not* claim improvements on Promptagator. We hope our findings will inform users on how to best maximize dense retriever effectiveness for domain-specific applications.

2 Promptagator

In the Promptagator setup, it is assumed that a large collection of n texts, denoted by $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$, is available but there are a limited number of annotated query-document pairs for training the

dense retrieval model. To overcome this challenge, Promptagator proposes to leverage an LLM's few-shot capabilities to amplify a *few* (e.g., 2-8) task-specific query-document pairs into a larger synthetic training dataset. A critical novelty of Promptagator, compared to similar work like InPars [2], is that it recognizes that retrieval tasks have different search intents (i.e., definitions of what relevance means). Thus, when generating domain-specific synthetic training data for a dense retriever, these search intents should be captured in the form of *few-shot prompting*.

The Promptagator workflow is as follows: Given d_i, a document in $\mathcal{D}$, an LLM is prompted to perform document-to-query generation, conditioned on *annotated*, task-specific query-document pairs, $\{(q_1, d_1), (q_2, d_2) \ldots (q_i, d_i)\}$, where q_i denotes an annotated query for document d_i. This process is repeated for k documents, where $k \leq n$, in turn generating many synthetic query-document pairs: $\{(\hat{q}_1, d_1), (\hat{q}_2, d_2) \ldots, (\hat{q}_k, d_k)\}$, where $\hat{q}$ denotes a synthetic query generated by the LLM. These synthetic query-document pairs are subsequently used to train a dense retrieval model.

While Promptagator demonstrated strong performance across various low-resource benchmarks, its results were based on a proprietary 137B FLAN-T5 [4] model, leaving open the question of whether smaller, accessible LLMs can serve as viable alternatives. In this paper, we study the effect of the latest open-source LLMs on performance in Promptagator-style dense retrieval model training.

3 Experimental Setup

We investigated various aspects of synthetic query generation and retrieval model training using select datasets from the BEIR benchmark [19] to train and evaluate our models. We used vLLM [14] for offline inference on various NVIDIA accelerators.

The selected datasets are detailed in Table 1. These were chosen to maintain a diversity of tasks, domains, and sizes while managing limited compute. For each dataset, we randomly sampled up to 100K documents to generate synthetic queries; we used the full corpus for datasets smaller than 100K documents.

Prompts were task-specific with few-shot examples, similar to those in Promptagator, and formatted in chat templates. This is an example system, user, and assistant prompt for SciFact:

```
System: You are a high-quality synthetic data generator. Your task
is to read an abstract from scientific research literature and
generate a relevant scientific claim. A claim is relevant if the
abstract contains all of the necessary evidence to support the
claim. Use the following examples to guide you. Respond with only
the scientific claim.
User: Abstract: {}
Assistant: {}
```

We selected few-shot examples from the development split when available, otherwise we used the test split (ArguAna, SCIDOCS, and SciFact only). To ensure our examples were relevant, we sampled from relevant queries with the highest scores. For instance, DBPedia [10] provides relevance assessment scores of 0 (irrelevant), 1 (relevant), and 2 (highly relevant); our few-shot examples all had a relevance assessment score of 2. For each example, the document text was injected into the content for the user role and the query text was injected into the content for the assistant role. We did not use document titles and removed the selected examples from the

set of documents used to generate synthetic queries.[2] We prompted an LLM with a task-specific system prompt with few-shot examples and a single document from which to generate a relevant query, and did so repeatedly for all datasets.

We also investigated two sets of sampling parameters (limited to those available in the vLLM library [14]). The first set came from Promptagator, which used a temperature of 0.7 across all models and model defaults elsewhere. The second set was selected to induce more diverse outputs and they are listed in Table 3. For all datasets, we generated a maximum of 256 tokens with 8 return sequences, meaning we generate up to 800K training queries for a given combination of LLM, sampling parameters, and dataset.

This study used open-source, transformer-based [21] decoder-only LLMs with varying model architectures and parameter sizes to generate synthetic queries. We refer to them as *QGen*. See Table 1 for the full list.

For training the dense retrieval model, we utilized the unsupervised Contriever [11] as the default backbone model. To train Contriever, we followed the setup from Dai et al. [6], using cross-entropy loss over in-batch random negatives. We trained with a learning rate of 2e-5 and batch size of 128 for datasets with less than 50K documents and a batch size 6000 otherwise. Prior to training, we randomly sampled 10% of our corpus as an evaluation set. Dense retrieval models were then trained for up to 30 epochs and the checkpoint with the lowest loss over the evaluation set was selected. We employed early stopping if the evaluation loss did not decrease for three consecutive epochs. Training was done using using GradCache [8] via Sentence Transformers [17] and retrieval experiments were performed with Pyserini [15]. We chose not to perform round-trip filtering as is done in Promptagator to better study the impact of the various QGen models.

4 Results

In this section we investigate the retrieval accuracy of Promptodile across different QGens and datasets. Results are shown in Table 1.

4.1 Promptodile versus Promptagator

When comparing Promptagator (no filtering) to Promptodile, we find that, across QGen models, Promptodile is generally on-par with Promptagator, even when using LLMs as small as 3B parameters. Besides Promptodile with gemma-3-1b, Promptodile always scores within one point — scoring between 43.7 to 46.2 NDCG@10 — of Promptagator (no filtering), which achieves an average of 44.7 NDCG@10. Additionally, we find that Promptodile with Phi-3-medium performs competitively with Promptagator (*with* filtering), suggesting that as LLMs improve, the filtering step may not be as important. For example, on DBPedia and HotpotQA — datasets where Promptagator gets the largest boost from round-trip filtering — the best Promptodile models (using Qwen2.5-14B and Phi-3-medium) score 2.5 and 3.4 points higher, respectively, than Promptagator.

4.2 Impact of LLM Scale on Promptodile

When comparing Promptodile within LLM families, we do not see strong improvements to Promptodile performance with QGens

---

[2]Since we removed all few-shot examples from the document sets, we did not remove them when evaluating as they are never exposed to the dense retrieval model.

Table 1: Main results (NDCG@10) across seven BEIR tasks. Note that for our experiments, we do *not* perform round-trip filtering to enable fair comparison across LLMs. As such, we also report Promptagator results without filtering, Promptagator (no filtering), based on Figure 2a in Dai et al. [6]. * denotes datasets in which Promptodile uses 10x less data than Promptagator.

Retriever	ArguAna	DBPedia*	FiQA	HotpotQA*	NFCorpus	SCIDOCS	SciFact	Avg.
BM25	39.7	31.8	23.6	63.3	32.2	14.9	67.9	39.1
Contriever	37.9	29.2	24.5	48.1	31.7	14.9	64.9	35.9
Contriever MS-MARCO	44.6	41.3	32.9	63.8	32.8	16.5	67.7	42.8
Promptagator (with filtering)	59.4	38.0	**46.2**	61.4	33.4	**18.4**	65.0	46.0
Promptagator (no filtering)	58.2	33.2	46.3	57.6	34.1	17.1	66.4	44.7
Promptodile (no filtering)								
QGen: Llama-3.2-3B	57.1	37.7	36.3	62.7	34.5	17.7	**69.9**	45.1
QGen: Llama-3.1-8B	59.4	35.4	37.7	62.1	34.1	16.7	69.9	45.0
QGen: Qwen2.5-3B	49.1	38.4	37.2	64.2	32.4	17.6	67.3	43.7
QGen: Qwen2.5-7B	59.8	37.3	35.9	64.1	33.4	16.1	64.7	44.5
QGen: Qwen2.5-14B	53.7	**40.5**	39.9	64.7	33.0	15.4	66.6	44.8
QGen: Phi-3-mini	56.1	37.6	36.5	62.6	34.7	18.2	69.6	45.0
QGen: Phi-3-medium	57.6	37.3	42.4	**65.0**	**35.2**	17.2	68.8	**46.2**
QGen: gemma-3-1b	54.4	31.2	20.9	59.0	32.9	12.1	66.4	39.6
QGen: gemma-3-4b	**61.8**	34.9	34.6	63.9	33.4	15.2	68.8	44.7
QGen: gemma-3-12b	58.7	35.1	37.6	63.1	33.9	11.7	65.8	43.7

Table 2: Impact of backbone dense retriever on NDCG@10. Promptodile is trained with Phi-3-medium Qgen model.

Retriever	ArguAna	NFCorpus	SCIDOCS	SciFact	Avg.
Contriever	37.9	31.7	14.9	64.9	37.4
Contriever MS-MARCO	44.6	32.8	16.5	67.7	40.4
E5 Base (Unsupervised)	42.2	35.8	21.1	73.7	43.2
E5 Base (Supervised)	51.4	36.6	19.0	73.1	45.0
Promptagator	59.4	33.4	18.4	65.0	44.1
Promptagator (no filtering)	58.2	34.1	17.1	66.4	44.0
Promptodile					
w/ Contriever	57.6	35.2	17.2	68.8	44.7
w/ E5 Base (Unsupervised)	57.5	**37.6**	**22.5**	**74.9**	**48.1**
w/ Contriever MS-MARCO	**59.4**	34.9	17.7	70.5	45.6

above 3B parameters. While gemma-3-1b shows the worst performance, scoring an average 39.6 versus the other Qgens which are between 43.7-46.2, we find that the improvement from using a 3B QGen versus a 14B QGen is minimal. In fact, within LLM families, on average Promptodile with 3B/4B parameter QGens are always within approximately one point of Promptodile with QGens of 7B+ parameters. These results demonstrate that smaller QGen models can effectively produce Promptagator-style dense retrieval systems, and there is no clear benefit to using larger, more expensive LLMs. The effectiveness of small QGen models makes fine-tuned dense retrieval models highly accessible.

To provide a cost comparison, we count the number of input and output tokens using one HotpotQA synthetic query dataset generated with Llama-3.2-3B as the QGen model[3]. In this dataset, we have ≈6.5M input tokens and ≈13.8M output tokens. At the time of writing, OpenAI charges $2/1M input tokens and $8/1M output tokens giving a total cost of $124[4] to generate synthetic queries

---
[3]Generated on 4xA100 GPUs (80GB VRAM) in ≈9 hours
[4]We ignore the cost of few-shot prompts which can be cached

for training. While not prohibitive on its own, this cost can grow quickly when factoring in many dozens of training datasets over time. In contrast, Llama-3.2-3B can easily be self-hosted.

4.3 Promptodile versus MS-MARCO Transfer

We compare Promptodile to Contriever MS-MARCO, and find that under the same backbone dense retrieval model (Contriever), Promptagator-style training with small, open-source LLMs is more effective than solely training on a large-scale dataset like MS-MARCO and *transferring* it to a new domain. This result further demonstrates the strong role LLMs can play as task-specific dataset generators. As we show in the next subsection, Promptodile and large-scale dataset training are not *independent* of each other, and can in fact be complementary.

4.4 Promptodile with Different Retrievers

Table 1 focused on Contriever as the backbone dense retriever for Promptodile as it shares similarities to the dense retriever used in the original Promptagator implementation. Here, we study how Promptodile can benefit using (1) an improved unsupervised dense retriever and (2) a dense retriever first trained on a large-scale dataset like MS-MARCO. To align with our goal of developing a solution for compute-constrained environments, we focus on BERT-base [7] sized models and leave exploration of larger dense retrievers as future work.

The results for this experiment are shown in Table 2. We find that fine-tuning Promptodile with E5 Base (Unsupervised) [23], a more effective unsupervised dense retriever than Contriever, can make strong improvements to the performance of Promptodile. Specifically, fine-tuning from E5 Base (Unsupervised) makes a 3.4 point boost, on average, upon fine-tuning from Contriever across the four datasets. When leveraging the same base dense retriever,

Table 3: Sampling parameters for more diverse queries.

temperature	top_p	top_k	repetition_penalty	presence_penalty
1.0	0.95	70	1.2	0.3

Figure 1: Difference in performance by sampling parameters, ($\text{NDCG@10}_{\text{default}} - \text{NDCG@10}_{\text{creative}}$)

Table 4: Relationship between verbosity and performance (using Spearman's rank correlation of NDCG@10 and normalized words per query).

dataset	ρ	p-value	n
FiQA	0.5758	0.0816	10
NFCorpus	0.6727	0.0233	11
SCIDOCS	0.6848	0.0289	10

but first fine-tuning on MS MARCO, Promptodile is able to improve by approximately 1 point on average, from 44.7 to 45.6 NDCG@10.

5 Synthetic Query Analysis

Dataset diversity is a key component in training performant LLMs [9, 25], and we hypothesize that generating more diverse or creative synthetic queries would also improve dense retrieval performance. We induce our QGen models to generate more creative outputs by changing sampling parameters for four datasets and our results are presented in Figure 1 with new parameters in Table 3. Generally, the impact is model-specific with Gemma and Qwen showing performance gains on more creative sampling parameters whereas Llama and Phi do better on default parameters. We see large swings in performance on ArguAna, highlighting the variation in performance one might expect on challenging tasks.

We then consider the importance of verbosity and measure it as the number of words per query, emphasizing words over tokens to facilitate comparison across different models' tokenizers. In this section, we investigate only QGen models with default sampling parameters. To adjust for varying lengths of documents (d), we normalize words per query (q) for 8 queries per document in each dataset (D) and take the average:

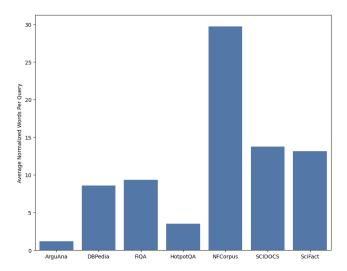$$\bar{r}_i = \frac{1}{8 * |D_i|} \sum_{d \in D_i} \sum_{q \in d} \frac{|d|}{|q|}$$

Figure 2: Verbosity by task.

where $\bar{r}_i$ is the average normalized length for the ith dataset. Documents are typically longer than queries, which we limit to 256 tokens, so we divide the length of the *document* by the length of the *query* for readability. This means verbosity increases as $\bar{r}_i$ decreases. As we can see in Figure 2, QGen models adapt well to various retrieval tasks, moderating the length of their queries depending on the given dataset. There is some support that verbosity can impact performance (see Table 4).

Finally, we use Meta-Llama-3.3-70B-Instruct as a judge [24] to score queries for ArguAna and NFCorpus on a scale of 0-3 as detailed by Thomas et al. [20]. Surprisingly, we find no overall correlation between relevance assessment scores (i.e., qrels) and NDCG@10 (standardized by dataset), but see a significant *negative* correlation with default sampling parameters (ρ=-0.4818, p-value=0.0270, n=21). We offer two hypotheses: (1) synthetic query-document pairs with higher average qrels have a higher rate of duplication. That is, individual queries are high-quality, but each set of eight queries are similar to one another, biasing average scores upward while overfitting, which is more pronounced on these smaller datasets (<10K documents each).[5] (2) Irrelevant synthetic queries are likely to overlap significantly with source documents since the queries are conditioned on those documents, exposing a limitation of LLMs as relevance assessors [1, 5]. We leave further investigation to future work.

6 Conclusion

In this work, we study how small-scale open-source LLMs can serve as query generators for Promptagator-style dense retriever training, which we refer to as *Promptodile*. Our results demonstrate that users can achieve competitive accuracy using LLMs as small as 3B parameters, making Promptagator-style fine-tuning of dense retrievers accessible for out-of-domain and low-compute settings.

---

[5]In contrast, creative sampling parameters using a higher temperature, and therefore considered more diverse, are positively correlated with performance with some confidence (ρ=0.3654, p-value=0.1131, n=20) [22].

GenAI Usage Disclosure

Generative AI tools were solely utilized to assist in making grammatical edits for sections of this work, including the text and tables.

References

[1] Marwah Alaofi, Paul Thomas, Falk Scholer, and Mark Sanderson. 2024. LLMs can be Fooled into Labelling a Document as Relevant: best café near me; this paper is perfectly relevant. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region (SIGIR-AP 2024)*. 32–41.

[2] Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. InPars: Unsupervised Dataset Generation for Information Retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022)*. 2387–2392.

[3] Leonid Boytsov, Preksha Patel, Vivek Sourabh, Riddhi Nisar, Sayani Kundu, Ramya Ramanathan, and Eric Nyberg. 2023. InPars-Light: Cost-Effective Unsupervised Training of Efficient Rankers. *arXiv preprint arXiv:2301.02998* (2023).

[4] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling Instruction-Finetuned Language Models. *Journal of Machine Learning Research* (2024), 1–53.

[5] Charles L.A. Clarke and Laura Dietz. 2024. LLM-based Relevance Assessment Still Can't Replace Human Relevance Assessment. *arXiv preprint arXiv:2412.17156* (2024).

[6] 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*.

[7] 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)*. 4171–4186.

[8] Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. 2021. Scaling Deep Contrastive Learning Batch Size under Memory Limited Setup. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*. 316–321.

[9] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783* (2024).

[10] Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. 2017. DBpedia-Entity v2: A Test Collection for Entity Search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017)*. 1265–1268.

[11] 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).

[12] Vitor Jeronymo, Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, Jakub Zavrel, and Rodrigo Nogueira. 2023. InPars-v2: Large Language Models as Efficient Dataset Generators for Information Retrieval. *arXiv preprint arXiv:2301.01820* (2023).

[13] 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)*. 6769–6781.

[14] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*. 611–626.

[15] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*. 2356–2362.

[16] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[17] 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 and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3982–3992.

[18] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval* (2009), 333–389.

[19] 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)*.

[20] Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2024. Large Language Models can Accurately Predict Searcher Preferences. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2024)*. 1930–1940.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *Advances in Neural Information Processing Systems* (2017).

[22] Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2022. GPL: Generative Pseudo Labeling for Unsupervised Domain Adaptation of Dense Retrieval. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2345–2360.

[23] 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 preprint arXiv:2212.03533* (2022).

[24] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. *Advances in Neural Information Processing Systems* (2023), 46595–46623.

[25] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. LIMA: Less Is More for Alignment. *Advances in Neural Information Processing Systems* (2023), 55006–55021.