# Leveraging LLMs for Unsupervised Dense Retriever Ranking

Ekaterina Khramtsova*
University of Queensland
St Lucia, Australia
e.khramtsova@uq.edu.au

Shengyao Zhuang*
CSIRO
Herston, Australia
shengyao.zhuang@csiro.au

Mahsa Baktashmotlagh
University of Queensland
St Lucia, Australia
m.baktashmotlagh@uq.edu.au

Guido Zuccon
University of Queensland
St Lucia, Australia
g.zuccon@uq.edu.au

## ABSTRACT

In this paper we present Large Language Model Assisted Retrieval Model Ranking (LARMOR), an effective unsupervised approach that leverages LLMs for selecting which dense retriever to use on a test corpus (target). Dense retriever selection is crucial for many IR applications that rely on using dense retrievers trained on public corpora to encode or search a new, private target corpus. This is because when confronted with domain shift, where the downstream corpora, domains, or tasks of the target corpus differ from the domain/task the dense retriever was trained on, its performance often drops. Furthermore, when the target corpus is unlabeled, e.g., in a zero-shot scenario, the direct evaluation of the model on the target corpus becomes unfeasible. Unsupervised selection of the most effective pre-trained dense retriever becomes then a crucial challenge. Current methods for dense retriever selection are insufficient in handling scenarios with domain shift.

Our proposed solution leverages LLMs to generate pseudo-relevant queries, labels and reference lists based on a set of documents sampled from the target corpus. Dense retrievers are then ranked based on their effectiveness on these generated pseudo-relevant signals. Notably, our method is the first approach that relies solely on the target corpus, eliminating the need for both training corpora and test labels. To evaluate the effectiveness of our method, we construct a large pool of state-of-the-art dense retrievers. The proposed approach outperforms existing baselines with respect to both dense retriever selection and ranking. We make our code and results publicly available at https://github.com/ielab/larmor/.

## CCS CONCEPTS

• **Information systems** → **Evaluation of retrieval results**.

## KEYWORDS

Model selection, Dense retrievers, Zero Shot Model Evaluation

**ACM Reference Format:**

Ekaterina Khramtsova, Shengyao Zhuang, Mahsa Baktashmotlagh, and Guido Zuccon. 2024. Leveraging LLMs for Unsupervised Dense Retriever Ranking. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3626772.3657798

*Equal Contribution


*SIGIR '24, July 14–18, 2024, Washington, DC, USA*

ACM ISBN 979-8-4007-0431-4/24/07.
https://doi.org/10.1145/3626772.3657798

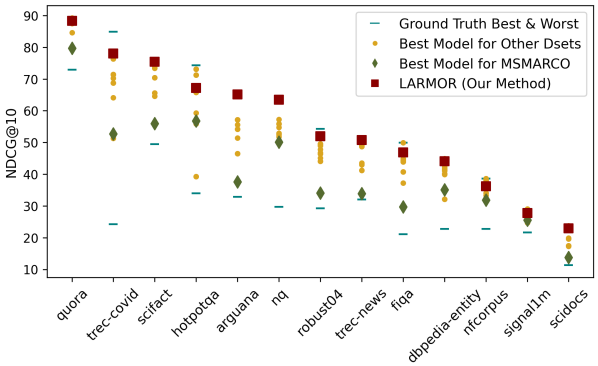


**Figure 1: nDCG@10 Across Collections. This figure illustrates that selecting the best model based on one collection (as indicated by orange dots) does not necessarily ensure its effectiveness on another. In contrast, our unsupervised approach (indicated by red squares) consistently selects competitive models across various collections, and even identifies the most performant model for Trec-News.**

## 1 INTRODUCTION

With the rapid advancements in natural language processing and information retrieval, a multitude of diverse dense retrievers (DRs) has emerged, demonstrating impressive effectiveness in various text retrieval tasks [56, 63]. For example, over 100 dense retrievers are featured in the Massive Text Embedding Benchmark[1] (MTEB [41]).

A typical situation in practical application settings is that a dense retriever trained on one or more corpora (the *training corpus*) is to be applied to a new corpus (the *target corpus*). Often queries and relevance judgements (labels) for the target corpus are not available, or they are prohibitive to collect due to costs or data access restrictions (this is often the case in domain specific settings like in health, legal and patent search). In this situation, then, search engine practitioners are faced with the question — Which dense retriever should I use? This is the task of *dense retriever selection* [28]: identify the most suitable DR for a target corpus. This is a challenging task because no queries and associated relevance judgements are available for the target corpus, and thus the prediction task is to be performed in an unsupervised manner.

A reasonable choice for dense retriever selection would be to select the DR that performs overall best on a comprehensive leaderboard like MTEB. However, recent studies have shown that the effectiveness of DRs is often dependent on the similarity between the training corpus and the target corpus; in particular, the effectiveness becomes varying and unpredictable when DRs are applied to data that differs from that at training (e.g., from a new domain, see Figure 1) [32, 35, 47, 54, 69, 70]. This issue is evident for instance in the MTEB benchmark. There, results show that the overall top-performing DRs may not necessarily be the most suitable for each

[1]https://huggingface.co/spaces/mteb/leaderboard

**Table 1: nDCG@10 on BEIR. 47 top-performing DRs from METB are used in this experiment. The first row (Oracle) reports the scores obtained when using the best DR for each collection, representing the upper bound score. The second row (Best DR) reports the scores of UAE, the DR that achieved the best average nDCG@10 on BEIR. The last row (LARMOR) reports the scores achieved using DRs selected by our method.**

| | NF | FiQA | ArguAna | SciDocs | SciFact | Covid | Quora | NQ | DBPedia | HotpotQA | Signal1M | Robust04 | Trec-News | **Avrg** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Oracle (Upper Bound) | 38.65 | 49.96 | 65.11 | 23.77 | 76.18 | 84.88 | 89.26 | 64.07 | 44.89 | 74.33 | 29.04 | 54.31 | 50.77 | 57.32 |
| Best DR (UAE) | 38.65 | 44.84 | 65.11 | 22.98 | 74.07 | 76.33 | 88.79 | 55.86 | 44.89 | 73.13 | 27.36 | 49.55 | 49.21 | 54.67 |
| LARMOR (ours) | 36.21 | 46.89 | 65.11 | 22.98 | 75.41 | 78.07 | 88.32 | 63.49 | 44.07 | 67.16 | 27.76 | 51.94 | 50.77 | 55.24 |

single collection. For instance, in our experiments with a subset of the DRs from the MTEB benchmark, `all-mpnet-base-v2`[2] is the top performing dense retriever on FiQA (nDCG@10 = 0.4996). In contrast, the overall leading dense retriever on the MTEB benchmark is `UAE-Large-V1`[3], which on FiQA exhibits a significant 10.3% loss in nDCG@10 compared to `all-mpnet-base-v2`.

Another straightforward approach to DR selection would be to select the DR that performs best on a held-out portion of the training corpus. This, has been shown to be the most effective method for selecting DRs in previous work [28]. However, a significant difficulty arises when doing this: new state-of-the-art DRs are often trained on multiple, proprietary, corpora, e.g., `e5` [58]. This renders access to training and/or held-out data impractical or impossible.

Other alternatives have been recently explored, adapted from similar problems in computer vision [28]. These, however, necessitate the availability of queries from the target corpus. This requirement poses a practical challenge in real-world scenarios, where the decision on which DR model to use must be made prior to deploying the application and thus often no prior logged queries are available. Nevertheless, even if logged queries are available, these approaches have been shown largely ineffective for DR selection [28].

In this paper, we propose a family of approaches for unsupervised, query-free dense retriever selection. At the core of these approaches is the leveraging of the capabilities of Large Language Models (LLMs) [64]. Specifically, we address the challenge posed by the absence of queries by using LLMs to generate synthetic queries for a (subset of the) target corpus. A document for which a synthetic query is generated and the generated query itself are considered forming a pseudo-relevant query-document pair for the target corpus. The set of pseudo-relevant query-document pairs are then used to estimate the ranking effectiveness of the DRs on the target corpus, and in turn this is used to rank DR systems.

Our results demonstrate that this straightforward performance estimation based on query generation is remarkably effective in selecting the most suitable DR for a target corpus – outperforming any other DR selection method. We further propose refinements to this idea that encompass the generation of synthetic relevance labels, and the exploitation of synthetic reference lists. The combination of these methods leads to a highly effective unsupervised strategy for dense retriever selection, which we refer to as Large Language Model Assisted Retrieval Model Ranking (LARMOR).

Table 1 provides a snapshot of LARMOR's predictive capabilities when selecting DRs for a target corpus: this serves as a motivation to delve further into the remainder of the paper. Each column in the table represents a target corpus (the last column is the mean effectiveness), and the value reported is the effectiveness on the target corpus of the selected dense retriever (fine-tuned on a different training corpus); Section 5 details our empirical settings. The first row (Oracle) refers to the best performance attainable if the effectiveness of every dense retriever on each of the target corpora were known – this is a theoretical upper bound. The second row reports the performance attainable when selecting the single model that performs overall best across all considered target corpora; in the case of the table, such a model is `UAE`. Again, this method is impossible in practice as it requires the true relevance labels for each target corpus to determine the overall best DR. Finally, the third row reports the remarkable performance of our LARMOR: these were obtained without resorting to human annotations, nor access to queries from the target corpus, which are often unfeasible to obtain before deployment. LARMOR in fact manages to select a highly competitive DR for each of the target corpora, and overall LARMOR provides better DR selection than using the `UAE` model across all target corpora – recall than `UAE` could have been selected only because we accessed the relevance labels of each target corpus. In addition, LARMOR performance is only 3.6% less than the theoretical upper bound (Oracle).

While our primary focus is on the DRs, our method can be applied to other IR models (e.g. re-rankers or sparse models). LARMOR is now integrated in the `DenseQuest` [27] system that implements DR selection over custom collections (https://densequest.ielab.io)

**Key contributions:**

(1) We introduce Large Language Model Assisted Retrieval Model Ranking (LARMOR), an approach for dense retriever selection that exploits the zero-shot capability of LLMs for generation of queries, relevance judgments, and reference lists. LARMOR is highly effective in selecting a dense retriever for a target corpus, without the need to supply queries or labels from the target corpus.
(2) To assess LARMOR's performance, we assemble a pool of 47 top DRs from the MTEB retrieval benchmark, extending the results of previous work to considering a consistently larger set of models.
(3) We conduct a thorough ablation study to examine factors that impact LARMOR's effectiveness, including the type and size of LLMs used, and the number of generated queries per documents.
(4) We augment the set of baselines for dense retriever selection by evaluating existing query performance prediction (QPP) methods overlooked by previous work [28].

## 2 RELATED WORK

### 2.1 Dense Retrievers Selection

While the task of unsupervised model selection has been widely studied for general deep learning models [8, 14, 19, 21, 29], its application in the context of neural rankers remains largely unexplored. The recent study by Khramtsova et al. [28] formalized the problem of DR selection and proposed several baseline methods. However,

[2]https://huggingface.co/sentence-transformers/all-mpnet-base-v2
[3]https://huggingface.co/WhereIsAI/UAE-Large-V1

their results indicated that most methods adapted from other areas are ineffective for IR. Differing from [28], we introduce a more challenging experimental setup by expanding the number of DRs and adding an additional constraint: the models might have different training sets. Consequently, we had to exclude several baselines, namely query similarity and Fréchet-based corpus similarity, as they require access to the training data. We retain the other baselines for our comparison—`MSMARCO perf`, `Binary entropy`, and `Query Alteration`—using the best-reported hyperparameters.

## 2.2 Query Performance Prediction

The concept of performance estimation in IR is primarily investigated within the context of query performance prediction (QPP) [7, 22, 23]. QPP aims to predict the retrieval quality of a search system for each query independently, without relying on relevance judgments. Our paper's objective differs slightly, focusing on comparing performance across different rankers, rather than evaluating each query within a single ranker. Nonetheless, it seems logical to explore the adaptation of QPPs to the task of DR selection.

Traditional QPP methods fall into three main categories [49]: those that assess the clarity of search results relative to the overall corpus; those that analyze the retrieval scores of documents within the ranking lists; and those that evaluate the robustness of the predicted ranking. We adapt methods from each of these categories to the DR selection task. From the first category, we employ `Clarity` [11]; from the second, we utilize `WIG` [65], `NQC` [50], `SMV` [53], and $\sigma$ [12, 43]; and from the third category, we explore `Fusion` [49]. Additionally, `Query Alteration` [28] can also be considered as an adaptation of robustness-based QPP. Our LARMOR method bears similarities with the concurrently proposed QPP-GenRE [39]; however QPP-GenRE is intended for the QPP task and it only considers producing synthetic labels, not queries.

## 2.3 Challenges For The Existing Baselines

Next we discuss the challenges encountered in adapting existing methods to our task of unsupervised DR ranking and selection.

*2.3.1 Normalizing factors for score-based methods.* Score-based QPP methods can be significantly enhanced by scaling their respective query-based measures with the relevance of the entire corpus to that query [31, 49]. In the context of DRs, this scaling is akin to the score between the query and the entire target corpus; however, calculating this value is computationally unfeasible [17]. Several methods have been proposed to approximate this scaling parameter. One approach, as suggested by Meng et.al. [38], is to take the average score of the top retrieved documents for each query as the normalizing factor. Another method, defined by Faggioli et.al. [17], involves representing the entire corpus as a centroid of its documents, derived from the latent space of the DR. The normalizing factor is then calculated as the score between the query and this centroid representation. We followed [38] in our experiments, and report both variations with and without normalization.

Typically, the scores generated for a query-document pair do not accurately reflect the probability of the query being relevant to the document. For example, in models trained using the dot product, the scores are not bound to a range between 0 and 1.

In addition to the variability of the scores within one collection, another challenge is the variability of the score distributions across DRs. This variability arises due to each dense retriever having unique architecture, loss function, and other training hyperparameters. As a result, even by normalizing the scores within one corpus, score-based methods do not perform well when used for comparing different dense retrievers, as will be shown in the next section.

*2.3.2 Variability of pre-trained tasks and training collections for performance-based methods.* DRs are seldom trained from scratch; rather, it is common to start with a model pre-trained on a different task and fine-tune it for a retrieval task. Consequently, it is logical to use the performance on the original task and the new retrieval task as indicators of model generalizability. For instance, if a model was pre-trained on a masked language modeling task (such as BERT [15] and Roberta [33]), one could evaluate the model's adaptability to a new task by examining its robustness to masking, as demonstrated in the Query Alteration Method. Similarly, if a model was fine-tuned on MSMARCO, its performance on MSMARCO can serve as an indicator of its ability to perform a retrieval task.

However, a challenge emerges because the dense retrievers in our study are not pre-trained using the same task. For instance, while some models are based on BERT, others are built on GPT [6] and were initially pre-trained using Next Token Prediction. Additionally, the training corpora differ across models. This diversity results in a performance evaluation that may be biased towards models trained on specific tasks or corpora, potentially not reflecting true performance on the target corpus or task.

## 2.4 LLMs for Information Retrieval

In our research, we rely on LLMs to generate pseudo-relevant queries, labels, and reference lists in the zero-shot setting. Numerous papers in the NLP and IR literature demonstrated the remarkable zero-shot performance of LLMs in these tasks.

For query generation, previous methods focused on fine-tuning pre-trained language models to generate pseudo-relevant queries [20, 42]. More recently, many works demonstrated that it is possible to generate high-quality queries for training ranking models [5, 13, 25, 57] by prompting LLMs in a zero-shot manner. Hence, following these works, we also use a domain-specific query generation prompt to guide LLMs in generating pseudo-relevant queries.

Another important component in our pipelines is the generation of pseudo-relevant judgments for the generated queries. Thomas et al. [55] demonstrate that using GPT-4 to generate relevant judgments can surpass human annotators in evaluating search systems. Similarly, a study conducted by Faggioli et al. [16] shows that LLM-generated relevant labels can align with labels annotated by TREC annotators. Thus, in this work, we leverage LLMs to provide additional relevance judgments in addition to the generated queries.

Finally, our work also leverages a pseudo-reference list to evaluate the effectiveness of DRs. This involves generating document rankings with high ranking scores (e.g., high nDCG). In IR, numerous works have demonstrated that LLMs have very strong zero-shot ranking capabilities. The methodologies for harnessing LLMs in zero-shot ranking tasks can be broadly categorized into pointwise [48, 66, 67], listwise [36, 44, 45, 51, 52], pairwise [46], and setwise [68]. In our paper, we adopt a setwise approach to generate a pseudo-reference list due to its high effectiveness and efficiency.

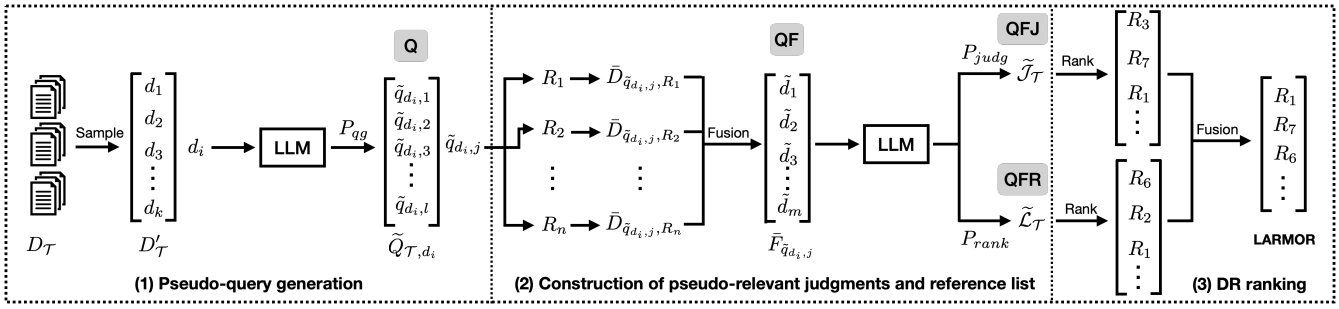


**Figure 2: The LARMOR dense retriever selection pipeline. Labels Q, QF, QFJ and QFR refer to the ablation points described in Section 6.2.**

## 3 PROBLEM FORMULATION

Let $\mathcal{T}$ be a target collection containing a corpus $D_{\mathcal{T}}$ of documents, a set $Q_{\mathcal{T}}$ of queries, and a set $\mathcal{J}_{Q_{\mathcal{T}},D_{\mathcal{T}}}$ of relevance judgments (i.e., labels), which reflect the degree of relevance of a given document $d \in D_{\mathcal{T}}$ in relation to a specific query $q \in Q_{\mathcal{T}}$. We note that accessing such relevance judgments presents significant challenges: queries often contain private user information; while collecting high-quality relevance judgments is time-consuming, and for some applications (e.g., medical or legal IR) requires in-depth domain knowledge and thus can be costly.

Let $\mathcal{R} = \{R_1, R_2, \dots R_n\}$ be a set of rankers, each trained on its respective training collection[4]: $\mathcal{S} = \{\mathcal{S}_{R_1}, \mathcal{S}_{R_2}, \dots, \mathcal{S}_{R_n}\}$. Note that the target collection $\mathcal{T}$ was not used during training of $\mathcal{R}$: $\mathcal{T} \notin \mathcal{S}$. In current applications, the training collections $\mathcal{S}_{R_i}$ often contain large corpora, which often include private documents – e.g., those used to train the latest state-of-the-art dense retrievers like e5 [58]. Therefore, we operate under the assumption that only the trained dense retrievers in $\mathcal{R}$ are available, while access to $\mathcal{S}$ is restricted.

Finally, let $\mathcal{E}$ be an evaluation measure, such as nDCG@10. Practitioners can establish an ordering of DRs in $\mathcal{R}$ based on the value of the evaluation measure $\mathcal{E}$ obtained on the target collection. This is achieved by applying each DR to the target collection and using the relevance judgments to compute the evaluation measure. The rankers are then arranged in decreasing order of $\mathcal{E}$, creating the ranking (ordering) of DRs $O(\mathcal{R}, \mathcal{T}, \mathcal{E}, \mathcal{J})$. The top-ranked DR is then typically selected for deployment as a search function on the target corpus of documents $D_{\mathcal{T}}$ to answer new queries, as it is the one that has been found performing best on the target collection $\mathcal{T}$. We note that we consider $O(\mathcal{R}, \mathcal{T}, \mathcal{E}, \mathcal{J})$ to be the ground truth ranking for the dense retriever selection task, defined below, since the DRs are evaluated and ranked based on the ground truth relevance judgments.

*Dense retriever selection task*: The problem of dense retriever selection consists of predicting the ranking $O(\mathcal{R}, \mathcal{T}, \mathcal{E}, \mathcal{J})$ without accessing the relevance judgments $\mathcal{J}_{Q_{\mathcal{T}},D_{\mathcal{T}}}$, as well as the target queries $Q_{\mathcal{T}}$. This is equivalent to producing a ranking $\hat{O}(\mathcal{R}, D_{\mathcal{T}}, \mathcal{E})$ of the rankers in $\mathcal{R}$ for the target collection $\mathcal{T}$ and evaluation measure $\mathcal{E}$, such that $\hat{O}(\mathcal{R}, D_{\mathcal{T}}, \mathcal{E})$ corresponds to the true ranking $O(\mathcal{R}, \mathcal{T}, \mathcal{E}, \mathcal{J})$. Note that $\hat{O}(\mathcal{R}, D_{\mathcal{T}}, \mathcal{E})$ does not include the relevance assessments $\mathcal{J}_{Q_{\mathcal{T}},D_{\mathcal{T}}}$ and target queries $Q_{\mathcal{T}}$ as input. Our goal is to develop a dense retriever selection method $M(\mathcal{R}, D_{\mathcal{T}})$ that produces the ranking $\hat{O}(\mathcal{R}, D_{\mathcal{T}}, \mathcal{E})$.

[4]Multiple training collections could also be used to derive a dense retriever: this does not change the setup discussed here.

## 4 METHODOLOGY

Next we describe our method, Large Language Model Assisted Retrieval Model Ranking (LARMOR). LARMOR uses Large Language Models (LLMs) to tackle the dense retriever selection problem. The method encapsulates a pipeline consisting on three crucial components, as outlined in Figure 2: (1) pseudo-query generation, (2) construction of pseudo-relevant judgments and reference lists, and (3) dense retrievers ranking.

### 4.1 Pseudo-Query Generation

The first step of the LARMOR pipeline involves generating pseudo-queries for the target corpus $D_{\mathcal{T}}$ to address the challenge of the absence (i.e., inability to access) of a representative set of queries for the target collection, $Q_{\mathcal{T}}$.

For this, we start by randomly sampling a subset of $k$ documents from the target corpus: $D'_{\mathcal{T}} = \{d_1, .., d_k\} \in D_{\mathcal{T}}$. Once the subset $D'_{\mathcal{T}}$ is built, each document from $D'_{\mathcal{T}}$ is passed to LARMOR's LLM, accompanied by a domain-specific query generation prompt $P_{qg}$, to generate a set of pseudo-relevant queries $\widetilde{Q}_{\mathcal{T}}$ specific to the target collection[5]. Note that for each sampled document $d_i$, we generate $l$ queries ($\tilde{q}_{d_i,1}, \dots \tilde{q}_{d_i,l}$):

$$\widetilde{Q}_{\mathcal{T}} = \bigcup_{i=1}^{k} \prod_{j=1}^{l} LLM(P_{qg}(d_i)). \tag{1}$$

in the equation, $\prod_l$ symbolises the generation of $l$ queries for a single document $d_i$, each time using the same prompt template $P_{qg}$ specific to the domain of the target collection.

For instance, in the case of the target domain being Wikipedia (e.g., for the NQ corpus in BEIR), the query generation prompt we employ is *"Generate a question that the following Wikipedia page can answer. Avoid generating general questions. Wikipedia page: $\{d_i\}$"*, where $d_i$ is a placeholder for the sample document text. By inputting this prompt to the LLM, it is possible to generate in-domain queries for the collection, thus addressing the challenge of the unavailability of a target query set $Q_{\mathcal{T}}$. This prompt design requires only minimal prior knowledge about the collection domain, effectively mimicking real-world scenarios. The design of prompts specific to a target collection has been shown effective in previous work in the context of training dense retrievers [3, 58, 62]. To the best of our knowledge, we are the first to design and use domain-specific prompts for query generation.

[5]That is, we assume that it is known what the representative search tasks for the target collection are.

It is worth noting that generating multiple pseudo-relevant queries for each sample document[6] using sampling generation strategies, such as top-p sampling [24], is reasonable. Since one document is likely to cover different topics, it could be relevant to various queries. Generating multiple queries with sampling strategies has the potential to cover different aspects of the document.

## 4.2 Construction of Pseudo Relevance Judgments and Reference Lists

Once $\widetilde{\mathcal{Q}}_{\mathcal{T}}$ is obtained, we could construct pseudo-relevant judgments to evaluate dense retrievers in the candidate pool by assuming a document is relevant to its corresponding generated queries. However, such an approach can only provide one relevant document per generated query, and these shallow relevance judgments may be sub-optimal for evaluating the ranking effectiveness of a DR [2, 34, 37, 40]. Consequently, the next step in the LARMOR pipeline is to construct comprehensive pseudo-relevant signals to evaluate and rank DRs in the candidate pool. We propose two different types of pseudo-relevant signals for this purpose: pseudo-relevant judgments and pseudo-reference lists, each with a distinct way of prompting LLMs and evaluating DRs.

For both pseudo-relevant signals types we start by creating a single document ranking for each generated query $\tilde{q}_{d_i,j} \in \widetilde{\mathcal{Q}}_{\mathcal{T},d_i}$, where $\widetilde{\mathcal{Q}}_{\mathcal{T},d_i}$ is the subset of $\widetilde{\mathcal{Q}}_{\mathcal{T}}$ that contains the $l$ generated queries for sampled document $d_i$. The single document ranking for the generate query $\tilde{q}_{d_i,j}$ is achieved by submitting the query to all considered DRs in $\mathcal{R} = \{R_1, R_2, ..., R_n\}$, to obtain the document rankings $\{\overline{D}_{\tilde{q}_{d_i,j},R_1}, \overline{D}_{\tilde{q}_{d_i,j},R_2}, ..., \overline{D}_{\tilde{q}_{d_i,j},R_n}\}$. Here, $\overline{D}_{\tilde{q}_{d_i,j},R_1}$ is the ranking of documents in the corpus $D_{\mathcal{T}}$ induced by dense retriever $R_1$ for query $\tilde{q}_{d_i,j}$. While the notation assumes that any $\overline{D}_{\tilde{q}_{d_i,j},R}$ is a total ordering of the documents in $D_{\mathcal{T}}$, in practice retrieval is conducted up to a rank cut-off (typically 1,000): the cut-off value has little effect on the result of the fusion, provided it is large enough (i.e. larger than parameter $m$ below, the number of documents selected from the fused ranking). Subsequently, a rank fusion algorithm[7] is employed to merge all the rankings for $\tilde{q}_{d_i,j}$, resulting in a single fused document ranking from which we select only the top-$m$ documents, obtaining the document ranking $\overline{F}_{\tilde{q}_{d_i,j}}$ of size $m$. One can consider this step as selecting the most valuable $m$ documents for query $\tilde{q}_{d_i,j}$ for the LLM to judge or rank: the documents in $\overline{F}_{\tilde{q}_{d_i,j}}$ are likely retrieved by most of the DRs. Hence, providing relevance judgments or reference lists for these documents may yield a more accurate evaluation of the DR effectiveness.

Next, $\overline{F}_{\tilde{q}_{d_i,j}}$ is passed as input to the LLM using either (or both) of two prompts (if both, this is done separately, i.e. independent inferences for each prompt).

Prompt $P_{judg}$ instructs the LLM to generate the pseudo-relevance judgments[8] $\widetilde{\mathcal{J}}_{\tilde{q}_{d_i,j},\overline{F}_{\tilde{q}_{d_i,j}}}$ for the m documents in $\overline{F}_{\tilde{q}_{d_i,j}}$:

$$\widetilde{\mathcal{J}}_{\tilde{q}_{d_i,j},\overline{F}_{\tilde{q}_{d_i,j}}} = \bigcup_{c=1}^{m} LLM(P_{judg}(\tilde{q}_{d_i,j}, \tilde{d}_c)). \quad (2)$$

where $\tilde{d}_c$ is a document in the fused ranking $\overline{F}_{\tilde{q}_{d_i,j}}$. We then can collate[9] pseudo-relevance judgements across all queries for a sample document, and all sample documents for the target corpus $D_{\mathcal{T}}$, obtaining the relevance judgements set $\widetilde{\mathcal{J}}_{\mathcal{T}}$ (or $\widetilde{\mathcal{J}}$ for brevity of notation, since we consider only one target collection $\mathcal{T}$).

To "implement" $P_{judg}$ we adapt the fine-grained relevance labels generation prompt of Zhuang et al. [66] because of its previously reported high effectiveness[10]. As an example, our relevance judgment prompt for the NQ collection is *"For the following query and document, judge whether they are 'Highly Relevant', 'Somewhat Relevant', or 'Not Relevant'. Query: {$\tilde{q}_j$} Document: {$\tilde{d}_c$}"*. Following Zhuang et al. [66], we only consider $\tilde{d}_c$ to be relevant to $\tilde{q}_j$ if the LLM generates *'Highly Relevant'*, i.e., we convert the graded judgement to binary judgments. Finally $\widetilde{\mathcal{J}}$ will be used for ranking DRs, which we discuss in details in the next section.

Prompt $P_{judg}$ guides the LLM to generate relevance judgments at a document level. In addition, we propose a second prompt, $P_{rank}$, to evaluate DRs at the ranking level. $P_{rank}$ is designed to instruct the LLM to generate a highly effective document ranking $\overline{L}_{\tilde{q}_{d_i,j}}$ to be used as pseudo-reference list for the generated query $\tilde{q}_{d_i,j}$. A reference list is commonly used in QPP [49], giving rise to effective predictive methods in that context. In our work, we adapt this idea to enhance our approach. This is achieved by prompting the LLM to re-rank the $m$ documents in the fused ranking for query $\tilde{q}_{d_i,j}$, i.e., $\overline{F}_{\tilde{q}_{d_i,j}}$:

$$\overline{L}_{\tilde{q}_{d_i,j}} = LLM(P_{rank}(\overline{F}_{\tilde{q}_{d_i,j}}) \quad (3)$$

To "implement" $P_{rank}$ we used the Setwise document ranking prompt [68]. We note other LLM ranking prompts could have been used, e.g., Pointwise [48, 66, 67], Listwise [36, 44, 45, 51, 52], Pairwise [46]. We chose the Setwise prompt because of its robustness and high effectiveness; due to space and computation constraints, we leave the study of other prompts for implementing $P_{rank}$ and their impact on LARMOR to future work.

Finally, we collate all reference lists into a set of reference list $\widetilde{\mathcal{L}}_{\mathcal{T}}$ (or for simplicity of notation, $\widetilde{\mathcal{L}}$) for the target corpus $\mathcal{T}$, keeping track of which generated query each list refers to.

## 4.3 Dense Retriever Ranking

The final step in our LARMOR pipeline is ranking all the dense retrievers in the candidate pool using either or both of the generated pseudo-relevance judgments $\widetilde{\mathcal{J}}$ and the pseudo-reference lists $\widetilde{\mathcal{L}}$.

To rank DRs for a target collection with the pseudo-relevance judgements, we first produce document rankings in answer to the generated queries using all the DRs, and we then evaluate each of these rankings using the target evaluation measure $\mathcal{E}$ (in our empirical evaluation, $\mathcal{E}$ = nDCG@10). We then average the evaluation values across all queries to associate an estimated average evaluation measure $\widetilde{\mathcal{E}}$ to each of the dense retrievers (it is estimated because pseudo queries and judgements are used, in place of the real ones from the target collection). Subsequently we rank DRs in descending order of $\widetilde{\mathcal{E}}$.

[6] Recall that we generate $l$ queries for each sample document.
[7] Which rank fusion method to use is an implementation choice; many exist [30].
[8] Sometimes referred to as synthetic judgements.
[9] Via the set union operation.
[10] We note other prompts for query-document relevance judgements have been proposed, e.g., that of Thomas et al. [55]; we adapted the one of Zhuang et al. [66] over others because of its simplicity. We leave evaluating the impact on LARMOR of alternative prompts for relevance evaluation to future work.

To rank DRs for a target collection with the reference lists $\widetilde{\mathcal{L}}$, we calculate the average Rank Bias Overlap (RBO) [59] of the rankings obtained using each DR for each generated query $\tilde{q}_{d_i,j}$ against its corresponding pseudo-reference list $\bar{L}_{\tilde{q}_{d_i,j}}$. We then rank DRs in descending order of RBO values.

Finally, as the above two rankings of dense retrievers have been obtained leveraging different relevance signals (judgements vs. reference lists), we posit it is beneficial to combine these rankings to obtain a comprehensive and effective ordering of the DRs, from the one thought to be most effective on the target corpus to the least effective. Thus, we further employ a fusion algorithm to merge the rankings of dense retrievers[11] and create our final ranking of DRs used as solution to the dense retrievers selection task.

# 5 EXPERIMENTAL SETUP

We aim to comprehensively evaluate our proposed DR selection approach against a large pool of state-of-the-art DRs across a wide range of corpora from different domains. In this section, we outline the details of the criteria for selecting DRs from the MTEB leaderboard, along with the corpora used in our experiments. Finally, we provide the implementation and evaluation details of our approach.

## 5.1 Dense Retriever Pool and Target Corpora

We assembled a large collection of state-of-the-art dense retrievers through the following steps:

We began by examining the MTEB leaderboard, selecting the top 30 retrievers based on their average performance across all corpora featured in the MTEB benchmark. Next, we assessed the performance of the retrievers on each individual corpus, expanding our set to include any retrievers that ranked in the top 30 for a specific corpus but were not part of our initial overall selection. This approach naturally led to overlapping models, as those most effective on one corpus often performed well on others. However, certain models demonstrated unique strengths in specific corpora. For instance, the `all-mpnet-base-v2` model is ranked the best for both SciDocs and FiQa corpora, yet it is 48th overall. Following the selection of leading models for each corpus, we refined our pool to align with our budgetary and computational constraints. This entailed removing API-based retrievers, e.g. Cohere, and any models with more than 6B parameters.

This process ultimately resulted in a carefully curated pool of 47 state-of-the-art dense retrieval models, which we will be using for all the experiments throughout the paper. Note that the number of models in our study substantially exceeds those utilized by Khramtsova et al. [28], thereby increasing the complexity of the task and enhancing the credibility of our results.

For evaluation corpora, we follow Khramtsova et al. [28] who utilized the corpora from the BEIR benchmark [54], which is widely employed for zero-shot dense retriever evaluation. This benchmark includes 18 collections across 9 diverse tasks. In accordance with standard practice, we selected a representative subset of 13 corpora, covering all 9 tasks featured in BEIR. The primary advantage of employing this benchmark is that none of its corpora were explicitly used for training the dense retrievers in our model pool. This makes it an appropriate choice for an unsupervised model selection task, especially in scenarios with domain shift between training and test.

## 5.2 Implementation Details

We employ LLMs in our proposed DR selection pipeline to generate queries, pseudo-relevance judgments, and pseudo-reference lists. We consistently use the FLAN-T5 [9] LLM through out the pipeline since it demonstrated strong effectiveness in zero-shot query generation [13] and document ranking [46, 67, 68].

For the query generation component, inspired by recent works of task-aware dense retriever training [3, 58, 62], we adapted their prompts to the task of query generation to generate in-domain pseudo-relevant queries for each BEIR corpus. Specifically, our query generation prompt templates have two key pieces of knowledge related to the target domain. Firstly, we specify the type of target query to generate, such as questions for question answering or arguments for argument retrieval. Secondly, we identify the type of document, distinguishing between sources like Wikipedia pages and scientific titles with abstracts. Due to space constraints, we direct readers to refer to our github repository for the list of our query generation prompts [12] and the resulting generated queries for each corpus [13]. We randomly sample $k = 100$ documents from the target corpus and employ top-p sampling with $p = 0.9$ to generate 10 queries for each sampled document, resulting in $|\widetilde{Q}_{\mathcal{T}}| = 1000$ generated queries per corpus. In Section 6.4, we investigate the impact of the number of generated queries per document as well as the influence of using different backbone LLMs in query generation.

Regarding the prompt for generating pseudo-relevance judgments, we modify the fine-grained relevance label generation prompts [66] to align with our domain-specific query generation prompts. This involves incorporating information about the query type and document type into the prompts. We then use the prompt to instruct LLMs to judge the top $m = 100$ documents from the $\overline{F}_{\tilde{q}_{d_i,j}}$ ranking for each generated query. We again refer readers to our github repository for the details of our prompts. As for the generation of pseudo-reference lists $\bar{L}_{\tilde{q}_{d_i,j}}$, we simply employ the original Setwise ranking prompt proposed by Zhuang et al. [68] with the default setting of using heap sort algorithm and compare 3 documents at a time to re-rank the top $m = 100$ documents from the $\overline{F}_{\tilde{q}_{d_i,j}}$.

For the fusion algorithm used in LARMOR to create $\overline{F}_{\tilde{q}_{d_i,j}}$ and the final DR ranking, considering that the scores provided by different DRs might have different scales, we opt for Reciprocal Rank Fusion (RRF)[10], a position-based method. We employ the implementation from the Python toolkit ranx[4] with the default parameters.

## 5.3 Evaluation

For evaluating our proposed LARMOR and baselines, we follow previous work that uses Kendall Tau correlation and $\Delta_e$ to assess the performance of methods on the DR selection task.

Both evaluations require the DR's ground truth performance ranking on the target corpus. Therefore, for each of the considered corpora, we rank the DRs based on the nDCG@10 obtained from the test queries and human judgments provided by each collection. This score is the official evaluation measure for BEIR.

[11] We further stress that in this step we fuse rankings of DRs, and not of documents like when creating reference lists in Section 4.2.

[12] https://github.com/ielab/larmor/blob/main/prompts.py

[13] https://github.com/ielab/larmor/tree/main/generated_data/

**Table 2: Kendall Tau Correlation value, calculated based on nDCG@10.**

| | NF | FiQA | ArguAna | SciDocs | SciFact | Covid | Quora | NQ | DBPedia | HotpotQA | Signal1M | Robust04 | Trec-News | **Avrg** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MSMARCO perf. | 0.337 | 0.240 | 0.17 | 0.118 | 0.291 | 0.339 | 0.298 | **0.646** | 0.515 | 0.492 | 0.121 | 0.141 | 0.186 | 0.300 |
| Binary Entropy | -0.056 | -0.164 | -0.048 | -0.086 | 0.103 | -0.183 | -0.280 | -0.081 | 0.034 | 0.165 | 0.069 | -0.106 | -0.212 | -0.065 |
| Query Alteration | -0.277 | -0.250 | -0.173 | -0.242 | -0.199 | -0.152 | -0.195 | -0.458 | -0.359 | -0.217 | 0.029 | -0.102 | -0.194 | -0.215 |
| WIG | -0.156 | -0.212 | -0.093 | -0.212 | -0.188 | -0.231 | -0.201 | -0.398 | -0.262 | -0.149 | 0.112 | -0.103 | -0.191 | -0.176 |
| WIG Norm | -0.027 | -0.125 | -0.138 | -0.105 | -0.001 | -0.118 | -0.147 | 0.078 | 0.053 | -0.049 | 0.010 | -0.106 | -0.055 | -0.056 |
| NQC | -0.036 | -0.021 | 0.142 | 0.08 | -0.01 | -0.202 | -0.071 | -0.304 | -0.153 | -0.086 | 0.036 | 0.051 | -0.006 | -0.045 |
| NQC Norm | -0.136 | -0.191 | -0.06 | -0.121 | -0.136 | -0.198 | -0.197 | -0.441 | -0.262 | -0.154 | 0.099 | -0.080 | -0.160 | -0.157 |
| SMV | -0.056 | -0.012 | 0.143 | 0.056 | 0.010 | -0.198 | -0.066 | -0.289 | -0.162 | -0.110 | 0.036 | 0.047 | 0.005 | -0.046 |
| SMV Norm | -0.173 | -0.179 | -0.066 | -0.136 | -0.127 | -0.204 | -0.191 | -0.429 | -0.280 | -0.197 | 0.103 | -0.075 | -0.154 | -0.162 |
| $\sigma$ | -0.204 | -0.228 | -0.116 | -0.149 | -0.186 | -0.198 | -0.142 | -0.402 | -0.260 | -0.219 | 0.084 | -0.099 | -0.167 | -0.176 |
| $\sigma_{max}$ | -0.147 | -0.236 | -0.123 | -0.114 | -0.182 | -0.224 | -0.236 | -0.370 | -0.291 | -0.166 | 0.062 | -0.064 | -0.227 | -0.178 |
| Clarity | 0.114 | 0.245 | 0.223 | 0.038 | 0.264 | -0.333 | 0.345 | 0.059 | -0.186 | -0.145 | 0.203 | 0.08 | -0.193 | 0.055 |
| Fusion | 0.653 | 0.436 | 0.544 | 0.686 | 0.636 | 0.368 | 0.670 | 0.374 | **0.719** | 0.555 | **0.506** | **0.611** | **0.698** | 0.574 |
| LARMOR (ours) | **0.700** | **0.618** | **0.627** | **0.739** | **0.766** | **0.553** | **0.740** | 0.563 | 0.665 | **0.710** | 0.380 | 0.444 | 0.690 | **0.630** |

After obtaining the ground truth DR performance ranking, Kendall Tau correlation is used to assess the similarity between the rankings generated by the DR selection methods and the ground truth ranking. Specifically, Kendall Tau correlation measures the proportion of document pairs that are ranked in the same order by both rankings. A score of 1 indicates perfect positive correlation, -1 indicates perfect negative correlation, and 0 indicates completely random correlation.

On the other hand, $\Delta_e$ aims to measure the performance gap between the selected DR and the ground truth best-performing DR for a specific DR evaluation measure $e$. This is defined as:

$$\Delta_e = e(M(\mathcal{R})) - e(R^*) \tag{4}$$

where $R^*$ is the ground truth best DR, and $M(\mathcal{R})$ is the DR ranked at top by method $M$. In our experiments, we set $e$ to be nDCG@10 to align with the target DR performance measurement. If $\Delta_e = 0$, it means that method $M$ successfully ranked the best DR at the top.

## 5.4 Other Baselines

We briefly describe the baselines used for comparison.

**Model Selection Methods [28]**:

- *MSMARCO performance*: ranks models based on their performance on MSMARCO - a large widely-used public collection for fine-tuning and evaluating retrieval models.
- *Binary Entropy* evaluates model uncertainty. For each query, the entropy of the probability-at-rank distribution is calculated. DRs are then ranked based on the average entropy across all queries.
- *Query Alteration* assesses the sensitivity of DRs to query variations. It involves modifying the query and measuring the standard deviation of scores for the retrieved documents. DRs are ranked based on the average standard deviation across queries, with smaller values indicating greater robustness against query perturbation, thereby implying higher retrieval credibility.

**QPP-based methods:**

- Weighted Information Gain (WIG) [65] measures the disparity between the average retrieval scores of top-ranked documents and the overall score of the corpus.
- Normalized Query Commitment (NQC) [50] calculates the standard deviation of the scores of top-ranked documents.
- Scores Magnitude and Variance (SMV) [53] combines both the magnitude and the standard deviation of the scores of top-ranked documents.
- Clarity [11] measures the discrepancy between the language model built from the top retrieved results and the language model of the entire corpus.
- $\sigma$ [43] calculates the standard deviation of scores, determining the optimal number of retrieved documents for each query to minimize the impact of low-scoring, non-relevant documents.
- $\sigma_{\max}$ [12] is a normalized standard deviation that considers only documents with scores above a certain percentage of the top score.
- Fusion [49] relies on submitting the target query to all candidate DRs to acquire document rankings for each DR. A search result fusion method is then used to aggregate these rankings into a pseudo-reference list. DRs are subsequently scored based on their RBO against this reference list.

For all QPP-based methods, the final score of a DR with respect to a target collection is computed as the average value returned by the QPP method across all queries.

# 6 RESULTS

## 6.1 Main Results

In Tables 2 and 3, we compare LARMOR against other baselines in terms of Kendall Tau and $\Delta_e$, respectively.

Firstly, we observe that the baseline `MSMARCO pref`, which simply ranks DRs based on nDCG@10 obtained on the MSMARCO collection, performs the best among the previous DR selection methods in terms of Kendall Tau; this finding aligns with previous work [28]. However, it is important to note that it is not guaranteed that MS MARCO training data is used in all the DRs we considered in our experiments. For example, `bge-large-en-v1.5` was trained on the Massive Text Pairs (MTP) collection, which contains 200M English text pairs [60]. It is the second-best performing DR across all collections, however it is predicted to be only the 8-th best if one relies on `MSMARCO pref`, being surpassed by DRs that likely overfit the MSMARCO collection but do not perform comparably well on the other collections. It is noteworthy that the prediction based on `MSMARCO pref` yields the best Kendall Tau for the NQ collection. This is somewhat expected since NQ is considered to be the most similar collection to MSMARCO.

Another performance-based approach, `Query Alteration`, closely follows `MSMARCO pref`. While it underperforms compared to `MSMARCO pref` in terms of Kendall Tau, it achieves a higher average $\Delta_e$, indicating its greater effectiveness in selecting the top DRs rather than providing the true ranking of all DRs.

**Table 3: $\Delta_e$, calculated based on nDCG@10.**

| | NF | FiQA | ArguAna | SciDocs | SciFact | Covid | Quora | NQ | DBPedia | HotpotQA | Signal1M | Robust04 | Trec-News | **Avrg** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MSMARCO Perf. | 6.84 | 20.21 | 27.47 | 10.02 | 20.26 | 32.18 | 9.58 | 13.91 | 9.80 | 17.52 | 3.51 | 20.23 | 16.89 | 16.03 |
| Binary Entropy | 6.84 | 18.54 | 25.84 | 11.59 | 25.06 | 9.10 | 2.21 | 34.34 | 6.26 | 29.2 | 5.48 | 7.59 | 12.31 | 14.95 |
| Query Alteration | 2.61 | 15.15 | 7.98 | 6.04 | 18.35 | 13.42 | 3.61 | 21.04 | 2.46 | 20.15 | 4.50 | 7.59 | 5.95 | 9.91 |
| WIG | 7.06 | 16.69 | 25.92 | 6.60 | 8.44 | 12.16 | 1.07 | 18.18 | 4.56 | 20.15 | 5.69 | 15.85 | 13.06 | 11.96 |
| WIG Norm | 13.46 | 24.25 | 24.45 | 12.42 | 22.39 | 14.58 | 3.98 | 17.36 | 13.38 | 29.74 | 1.82 | 9.22 | 14.82 | 15.53 |
| NQC | 15.09 | 20.48 | 23.22 | 11.59 | 25.05 | 19.59 | 4.04 | 19.68 | 16.79 | 29.2 | 4.14 | 15.28 | 12.31 | 16.65 |
| NQC Norm | 7.06 | 16.69 | 25.92 | 6.60 | 8.44 | 15.17 | 4.29 | 7.77 | 13.34 | 20.15 | 5.69 | 15.85 | 13.06 | 12.32 |
| SMV | 15.09 | 20.48 | 23.22 | 11.59 | 25.05 | 19.59 | 4.04 | 19.68 | 16.79 | 29.2 | 4.14 | 15.28 | 12.31 | 16.65 |
| SMV Norm | 7.61 | 15.15 | 25.92 | 9.83 | 8.44 | 15.17 | 4.29 | 7.77 | 13.34 | 7.73 | 5.69 | 15.85 | 13.06 | 11.53 |
| $\sigma$ | 7.61 | 16.69 | 25.92 | 6.60 | 10.58 | 15.17 | 4.29 | 4.96 | 3.60 | 20.15 | 4.50 | 15.85 | 10.36 | 11.25 |
| $\sigma_{max}$ | 7.61 | 16.69 | 11.45 | 6.67 | 5.74 | 15.17 | 2.20 | 21.04 | 6.26 | 18.64 | 3.62 | 15.85 | 10.36 | 10.87 |
| Clarity | 2.44 | 12.46 | 18.59 | 2.12 | 11.62 | 33.56 | 3.52 | 4.06 | 12.8 | 35.04 | 2.00 | 23.43 | 7.84 | 13.04 |
| Fusion | **1.15** | 5.46 | 7.96 | 5.09 | 3.67 | 20.09 | **0.94** | 9.29 | 2.53 | **7.17** | **0.53** | **2.37** | **0.0** | 5.10 |
| LARMOR (ours) | 2.44 | **3.07** | **0.0** | **0.79** | **0.77** | **6.81** | **0.94** | **0.58** | **0.82** | **7.17** | 1.28 | **2.37** | **0.0** | **2.08** |

**Table 4: Ablation Study: the effect of different steps of the pipeline. Kendall Tau Correlation value, calculated based on nDCG@10.**

| | NF | FiQA | ArguAna | SciDocs | SciFact | Covid | Quora | NQ | DBPedia | HotpotQA | Signal1M | Robust04 | Trec-News | **Avrg** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Q | 0.483 | 0.525 | 0.522 | 0.545 | 0.708 | 0.563 | 0.635 | 0.578 | 0.658 | 0.824 | 0.153 | 0.324 | 0.634 | 0.550 |
| QF | 0.677 | 0.545 | 0.502 | 0.729 | 0.610 | 0.313 | 0.622 | 0.311 | 0.648 | 0.520 | 0.517 | 0.539 | 0.648 | 0.552 |
| QFJ | 0.552 | 0.562 | 0.522 | 0.59 | 0.809 | 0.559 | 0.677 | 0.646 | 0.667 | 0.789 | 0.191 | 0.397 | 0.648 | 0.585 |
| QFR | 0.700 | 0.562 | 0.448 | 0.764 | 0.676 | 0.370 | 0.672 | 0.444 | 0.613 | 0.607 | 0.526 | 0.437 | 0.667 | 0.576 |
| LARMOR | 0.700 | 0.618 | 0.627 | 0.739 | 0.766 | 0.553 | 0.740 | 0.563 | 0.665 | 0.710 | 0.380 | 0.444 | 0.690 | 0.630 |

**Table 5: Ablation Study: the effect of different steps of the pipeline. $\Delta_e$, calculated based on nDCG@10.**

| | NF | FiQA | ArguAna | SciDocs | SciFact | Covid | Quora | NQ | DBPedia | HotpotQA | Signal1M | Robust04 | Trec-News | **Avrg** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Q | 1.52 | 5.46 | 13.73 | 3.26 | 2.01 | 18.29 | 0.0 | 0.58 | 3.87 | 3.10 | 1.38 | 7.92 | 5.95 | 5.16 |
| QF | 0.22 | 5.46 | 7.96 | 0.79 | 3.67 | 18.31 | 0.94 | 9.29 | 2.53 | 7.17 | 0.25 | 2.37 | 0.0 | 4.53 |
| QFJ | 1.74 | 3.07 | 1.35 | 3.90 | 0.77 | 5.31 | 0.34 | 0.58 | 4.56 | 3.10 | 1.38 | 7.92 | 5.95 | 3.07 |
| QFR | 0.22 | 5.46 | 8.08 | 0.79 | 0.77 | 14.67 | 0.94 | 8.21 | 3.73 | 7.17 | 0.25 | 2.37 | 0.0 | 4.05 |
| LARMOR | 2.44 | 3.07 | 0.0 | 0.79 | 0.77 | 6.81 | 0.94 | 0.58 | 0.82 | 7.17 | 1.28 | 2.37 | 0.0 | 2.08 |

As expected, methods that rely on comparing the scores produced by the retrievers perform poorly in both DR selection and DR ranking tasks. These methods include `Binary Entropy` and four QPP methods (WIG, NQC, SMV, $\sigma$). Notably, the normalized versions of score-based QPPs yield better results in terms of both $\Delta_e$ and Kendall Tau (Tables 2 and 3). This implies that incorporating a scaling parameter for normalizing scores is beneficial. However, as discussed in Section 2.3, this normalization primarily regularizes scores within the collection, but does not address the challenge of score distribution diversity across different DRs.

In contrast, `Fusion`, which aggregates the retrieved lists from different DRs without relying on absolute score values, achieves significantly better performance in terms of both Kendall Tau and $\Delta_e$. Nevertheless, similar to the other QPP baselines, `Fusion` requires the availability of the queries from the target collection. This requirement is often impractical, as DR selection must occur before the system is deployed and queries are gathered. It is important to note that our LARMOR, unlike other baselines, is query-free.

Finally, our LARMOR achieves the best overall performance in terms of both Kendall Tau and $\Delta_e$. Moreover, LARMOR selects the best DR for two collections (Arguana and Trec-News), resulting in an average nDCG drop of only 2.08% across the board.

In the next subsections we tease out the contributions of LARMOR's components, and the effect of the LLM model size, of the specific LLM backbone, and of the number of generated queries.

## 6.2 Effect of LARMOR's Components

We study the effectiveness of some of LARMOR's components, measuring effectiveness at intermediate ablation points in the pipeline, illustrated in Figure 2. The ablation points we investigate are:

- **Q:** Query generation — we use the LLM-generated query, along with its associated document, as a pseudo-relevant query-document pair. The DRs are then evaluated and ranked based on these judgments. Note that here we only have a single relevant document for each generated query.
- **QF:** Rank fusion of the generated queries — we generate multiple queries for each sampled document $d_i$, and fuse their ranking to obtain $\overline{F}_{\tilde{q}_{d_i,j}}$ for each $d_i$. We then rank DRs based on the average RBO value against the obtained set of fused rankings $\overline{F}_{\tilde{q}_{d_i,j}}$.
- **QFJ:** Pseudo-judgments from rank fusion — we generate multiple queries for each sampled document $d_i$, and fuse their ranking to obtain $\overline{F}_{\tilde{q}_{d_i,j}}$ for each $d_i$. We then use the LLM to generate pseudo-judgements ($\widetilde{\mathcal{J}}$), and use these to rank DRs.
- **QFR:** Pseudo-reference lists from rank fusion — using the same $\overline{F}_{\tilde{q}_{d_i,j}}$ described above, we use the LLM to re-rank $\overline{F}_{\tilde{q}_{d_i,j}}$, obtaining the pseudo-reference list $\overline{L}_{\tilde{q}_{d_i,j}}$. We then rank DRs with respect to the set of all reference lists $\mathcal{L}$.

The full LARMOR pipeline performs an additional fusion of the ranking of DRs obtained from the ablation points QFJ and QFR.

In Table 4 and 5, we present Kendall Tau and $\Delta_e$ result, obtained at these different steps of our pipeline.

As the results illustrate, with just judgments from Q, we can already achieve very strong performance that, outperforming most baselines, only falling short of QPP fusion.

QF further improves Q by using the fused ranking and RBO to rank DRs. We note that QF is similar to the QPP `Fusion` baseline except that the queries for QF are LLM-generated, while the queries for `Fusion` are the actual test queries. Remarkably, QF can surpass

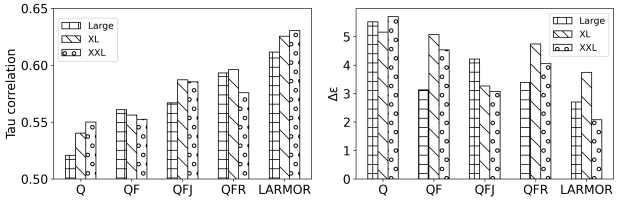


**Figure 3: Kendall Tau (left) and $\Delta_e$ (right) performance of our proposed LARMOR and its various components using different sizes of FlanT5 models.**

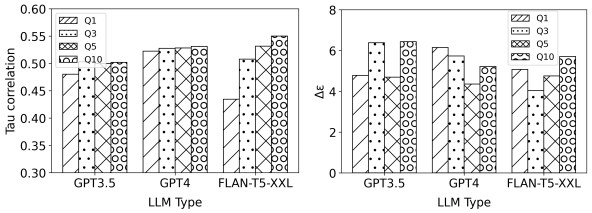


**Figure 4: Kendall Tau (left) and $\Delta_e$ (right) performance of pseudo-query generation with a different number of generated queries and different LLM backbone.**

QPP `Fusion` in terms of $\Delta_e$ average score, suggesting that our LLM-generated in-domain queries are of satisfactory quality.

On the other hand, QFJ and QFR prove to be very important; they both significantly improve QF for both Kendall Tau and $\Delta_e$.

Finally, our whole pipeline LARMOR achieved the overall best performance by fusing QFJ and QFR. These results demonstrate that each component in LARMOR has a significant contribution.

## 6.3 Effect of LLM Model Size

To comprehensively understand the impact of LLM size on our proposed LARMOR, in Figure 3 we plot the Kendall Tau and $\Delta_e$ performance across different steps of the pipeline. We explore the influence of FlanT5 models with varying sizes, namely FLAN-T5-large (780M), FLAN-T5-XL (3B), and FLAN-T5-XXL (11B).

Firstly, a clear pattern emerges as each step within the pipeline consistently contributes to improved effectiveness, irrespective of the model size with the only exception that FLAN-T5-large exhibits a suboptimal $\Delta_e$ score on QF compared to Q.

On the other hand, the performance across different model sizes exhibits variations at each pipeline step. This suggests that the conventional scaling law of LLMs might not apply here: it is not always the case that a larger model performs better for our method with FLAN-T5 models. Nevertheless, FLAN-T5-XXL achieved the best performance when the full LARMOR pipeline is applied.

## 6.4 Effect of LLM Backbone and Number of Generated Queries

In this section, we study how the state-of-the-art OpenAI LLMs, GPT-3.5 and GPT-4, perform compared to FLAN-T5-XXL. For these experiments, we only conduct tests on the query generation (Q) step due to the high cost of running the whole LARMOR with OpenAI models. Additionally, we investigate how the number of generated queries per document impacts performance. The results are illustrated in Figure 4.

For Kendall Tau performance, it is evident that more generated queries per document tend to be beneficial, especially for FlanT5-XXL, although the improvements are marginal for GPT-3.5 and GPT-4. GPT-4 tends to have an overall high Kendall Tau score; however, FLAN-T5-XXL only outperforms when the number of generated queries is set to 10. As for the $\Delta_e$ score, the impact of the number of generated queries varies, and the difference between models also varies.

However, we note that, although overall OpenAI models perform similarly to FLAN-T5-XXL, on the Arguana collection where FLAN-T5-XXL performs poorly (Kendall Tau = 0.522, $\Delta_e$ = 13.73), OpenAI models achieved surprisingly good performance. For example, GPT-4 achieved Kendall Tau = 0.713 and $\Delta_e$ = 0, which are the best scores on this collection. We observe that Arguana poses a non-trivial retrieval task—specifically, a counter-arguments retrieval task—requiring LLMs to have the capability to generate counter-arguments. GPT models demonstrate this capability effectively.

## 7 CONCLUSION

This paper introduces a novel LLM-based approach for dense retriever selection, the Large Language Model Assisted Retrieval Model Ranking (LARMOR). Dense retriever selection is a crucial task in many applications of search engines technologies. Dense retrievers are an effective and increasingly popular component of a search engine. Search engine practitioners are often faced with the choice of which dense retriever to deploy on a specific target collection. However, it is challenging to predict a DR's effectiveness on a target collection that contains data different from that in the collection used for training the DR. This is even more so if the practitioners do not have access to user queries and relevance judgements from the target collection, as it is often the case in many applications, e.g., in small-medium enterprises and in domains like health and legal, due to the cost and time required to collect these signals, or the impossibility to access this data for privacy reasons.

Notably, LARMOR stands out as the only available method that does not require any post-deployment data but instead relies on minimal prior knowledge about the target collection to design prompts to guide LLMs in generating synthetic queries, pseudo-relevant judgments, and reference lists. These in turn are used within LARMOR to rank dense retriever systems.

We comprehensively evaluate LARMOR across 13 different BEIR collections, considering a large pool of state-of-the-art dense retrievers. Our results demonstrate that LARMOR accurately ranks DRs based on their effectiveness in a zero-shot manner, outperforming all previous DR selection methods and adapted QPP methods.

Notably, unlike many existing baselines (e.g., score-based QPP, `Query Alteration`, `Binary Entropy`), our method is model-agnostic and can be extended to choose among any type of IR models.

For future work, we are interested in applying advanced automatic prompt optimization methods [18, 61] to further enhance the domain-specific prompts used by LARMOR. Additionally, we are also interested in incorporating into LARMOR and study recent advanced open-source LLMs, such as Mistral [26] and Llama3 [1]. Another promising avenue for future work is reducing the computational overhead of LLM-related computations in our pipeline.

# REFERENCES


[1] AI@Meta. 2024. Llama 3 Model Card. (2024). https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

[2] Negar Arabzadeh, Alexandra Vtyurina, Xinyi Yan, and Charles L. A. Clarke. 2022. Shallow pooling for sparse labels. *Information Retrieval Journal* 25, 4 (2022), 365–385. https://doi.org/10.1007/s10791-022-09411-0

[3] Akari Asai, Timo Schick, Patrick Lewis, Xilun Chen, Gautier Izacard, Sebastian Riedel, Hannaneh Hajishirzi, and Wen-tau Yih. 2023. Task-aware Retrieval with Instructions. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, 3650–3675. https://doi.org/10.18653/v1/2023.findings-acl.225

[4] Elias Bassani. 2022. ranx: A Blazing-Fast Python Library for Ranking Evaluation and Comparison. In *Proceedings of the 46th European Conference on Information Retrieval (ECIR)*, Vol. 13186. Springer, 259–264. https://doi.org/10.1007/978-3-030-99739-7_30

[5] 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 '22)*. Association for Computing Machinery, 2387–2392. https://doi.org/10.1145/3477495.3531863

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20)*. Curran Associates Inc., Article 159, 25 pages.

[7] David Carmel and Oren Kurland. 2012. Query performance prediction for IR. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '12)*. 1196–1197.

[8] Mayee Chen, Karan Goel, Nimit Sohoni, Fait Poms, Kayvon Fatahalian, and Christopher Re. 2021. Mandoline: Model Evaluation under Distribution Shift. In *Proceedings of the 38th International Conference on Machine Learning (ICML' 2021)*.

[9] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416* (2022).

[10] Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)*. ACM, 758–759.

[11] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. 2002. Predicting query performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '02)*. Association for Computing Machinery, 299–306. https://doi.org/10.1145/564376.564429

[12] Ronan Cummins, Joemon Jose, and Colm O'Riordan. 2011. Improved query performance prediction using standard deviation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '11)*. Association for Computing Machinery, 1089–1090. https://doi.org/10.1145/2009916.2010063

[13] Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot Dense Retrieval From 8 Examples. In *Proceedings of the 10th International Conference on Learning Representations (ICLR '22)*. https://openreview.net/forum?id=gmL46YMpu2J

[14] Weijian Deng and Liang Zheng. 2021. Are Labels Always Necessary for Classifier Accuracy Evaluation?. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '20)*.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[16] Guglielmo Faggioli, Laura Dietz, Charles L. A. Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, and Henning Wachsmuth. 2023. Perspectives on Large Language Models for Relevance Judgment. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR '23)*. Association for Computing Machinery, 39–50. https://doi.org/10.1145/3578337.3605136

[17] Guglielmo Faggioli, Thibault Formal, Simon Lupart, Stefano Marchesin, Stephane Clinchant, Nicola Ferro, and Benjamin Piwowarski. 2023. Towards Query Performance Prediction for Neural Information Retrieval: Challenges and Opportunities. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR '23)*. Association for Computing Machinery, 51–63. https://doi.org/10.1145/3578337.3605142

[18] Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2023. Promptbreeder: Self-Referential Self-Improvement Via Prompt Evolution. *arXiv preprint arXiv:2309.16797* (2023).

[19] Saurabh Garg, Sivaraman Balakrishnan, Zachary Chase Lipton, Behnam Neyshabur, and Hanie Sedghi. 2022. Leveraging Unlabeled Data to Predict Out-of-Distribution Performance. In *Proceedings of the 10th International Conference on Learning Representations (ICLR '22)*. https://arxiv.org/abs/2201.04234

[20] Mitko Gospodinov, Sean MacAvaney, and Craig Macdonald. 2023. Doc2Query–: When Less is More. In *Proceedings of the 45th European Conference on Information Retrieval (ECIR '23)*. Springer, 414–422.

[21] Devin Guillory, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, and Ludwig Schmidt. 2021. Predicting with Confidence on Unseen Distributions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV '21)*. 1114–1124.

[22] Claudia Hauff, Djoerd Hiemstra, and Franciska de Jong. 2008. A survey of pre-retrieval query performance predictors. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08)*. 1419–1420.

[23] Ben He and Iadh Ounis. 2006. Query performance prediction. *Information Systems* 31, 7 (2006), 585–594.

[24] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *Proceedings of the 8th International Conference on Learning Representations (ICLR '20)*.

[25] 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. https://doi.org/10.48550/ARXIV.2301.01820

[26] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825* (2023).

[27] Ekaterina Khramtsova, Teerapong Leelanupab, Shengyao Zhuang, Mahsa Baktashmotlagh, and Guido Zuccon. 2024. Embark on DenseQuest: A System for Selecting the Best Dense Retriever for a Custom Collection. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*. Association for Computing Machinery, New York, NY, USA (To Appear).

[28] Ekaterina Khramtsova, Shengyao Zhuang, Mahsa Baktashmotlagh, Xi Wang, and Guido Zuccon. 2023. Selecting which Dense Retriever to use for Zero-Shot Search. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region (SIGIR-AP '23)*. Association for Computing Machinery, 223–233. https://doi.org/10.1145/3624918.3625330

[29] Ekaterina Khramtsova, Guido Zuccon, Xi Wang, and Mahsa Baktashmotlagh. 2023. Convolutional Persistence as a Remedy to Neural Model Analysis. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics (AISTATs '23)*, Vol. 206. PMLR, 10839–10855. https://proceedings.mlr.press/v206/khramtsova23a.html

[30] Oren Kurland and J Shane Culpepper. 2018. Fusion in information retrieval. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '18)*. 1383–1386.

[31] Oren Kurland, Anna Shtok, Shay Hummel, Fiana Raiber, David Carmel, and Ofri Rom. 2012. Back to the roots: a probabilistic framework for query-performance prediction. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12)*. Association for Computing Machinery, 823–832. https://doi.org/10.1145/2396761.2396866

[32] Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to Train Your DRAGON: Diverse Augmentation Towards Generalizable Dense Retrieval. *arXiv preprint arXiv:2302.07452* (2023).

[33] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[34] Xiaolu Lu, Alistair Moffat, and J Shane Culpepper. 2016. The effect of pooling and evaluation depth on IR metrics. *Information Retrieval Journal* 19, 4 (2016), 416–445.

[35] Simon Lupart, Thibault Formal, and Stéphane Clinchant. 2022. MS-Shift: An Analysis of MS MARCO Distribution Shifts on Neural Retrieval. In *Proceedings of the 44th European Conference on Information Retrieval (ECIR '22)*. https://api.semanticscholar.org/CorpusID:256231516

[36] 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* (2023).

[37] Joel Mackenzie, Matthias Petri, and Alistair Moffat. 2021. A sensitivity analysis of the MSMARCO passage collection. *arXiv preprint arXiv:2112.03396* (2021).

[38] Chuan Meng, Negar Arabzadeh, Mohammad Aliannejadi, and Maarten de Rijke. 2023. Query Performance Prediction: From Ad-hoc to Conversational Search. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. Association for Computing Machinery, 2583–2593. https://doi.org/10.1145/3539618.3591919

[39] Chuan Meng, Negar Arabzadeh, Arian Askari, Mohammad Aliannejadi, and Maarten de Rijke. 2024. Query Performance Prediction using Relevance Judgments Generated by Large Language Models. *arXiv preprint arXiv:2404.01012* (2024).

[40] Alistair Moffat, Falk Scholer, and Ziying Yang. 2018. Estimating Measurement Uncertainty for Information Retrieval Effectiveness Metrics. *J. Data and Information Quality* 10, 3 (2018), 22. https://doi.org/10.1145/3239572

[41] Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive Text Embedding Benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2014–2037. https://doi.org/10.18653/v1/2023.eacl-main.148

[42] Rodrigo Nogueira and Jimmy Lin. 2019. From doc2query to docTTTTTquery. https://api.semanticscholar.org/CorpusID:208612557

[43] Joaquín Pérez-Iglesias and Lourdes Araujo. 2010. Standard deviation as a query hardness estimator. In *Proceedings of the 17th International Conference on String Processing and Information Retrieval (SPIRE'10)*. Springer-Verlag, 207–212.

[44] Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. RankVicuna: Zero-Shot Listwise Document Reranking with Open-Source Large Language Models. *arXiv preprint arXiv:2309.15088* (2023).

[45] Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. RankZephyr: Effective and Robust Zero-Shot Listwise Reranking is a Breeze! *arXiv preprint arXiv:2312.02724* (2023).

[46] 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* (2023).

[47] Ruiyang Ren, Yingqi Qu, Jing Liu, Xin Zhao, Qifei Wu, Yuchen Ding, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2023. A Thorough Examination on Zero-shot Dense Retrieval. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP '23)*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, 15783–15796. https://doi.org/10.18653/v1/2023.findings-emnlp.1057

[48] 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 (EMNLP '22)*. Association for Computational Linguistics, 3781–3797. https://doi.org/10.18653/v1/2022.emnlp-main.249

[49] Anna Shtok, Oren Kurland, and David Carmel. 2016. Query Performance Prediction Using Reference Lists. *ACM Transactions on Information Systems (TOIS)* 34 (2016), 1 – 34. https://api.semanticscholar.org/CorpusID:14981277

[50] Anna Shtok, Oren Kurland, David Carmel, Fiana Raiber, and Gad Markovits. 2012. Predicting Query Performance by Query-Drift Estimation. *ACM Transactions on Information Systems* 30 (2012). https://doi.org/10.1145/2180868.2180873

[51] 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 (EMNLP '23)*. Association for Computational Linguistics, 14918–14937. https://doi.org/10.18653/v1/2023.emnlp-main.923

[52] Manveer Singh Tamber, Ronak Pradeep, and Jimmy Lin. 2023. Scaling Down, LiTting Up: Efficient Zero-Shot Listwise Reranking with Seq2seq Encoder-Decoder Models. *arXiv preprint arXiv:2312.16098* (2023).

[53] Yongquan Tao and Shengli Wu. 2014. Query Performance Prediction By Considering Score Magnitude and Variance Together. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM '14)*. Association for Computing Machinery, 1891–1894. https://doi.org/10.1145/2661829.2661906

[54] 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 *Proceedings of the 35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS '21)*.

[55] Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2023. Large language models can accurately predict searcher preferences. *arXiv preprint arXiv:2309.10621* (2023).

[56] Nicola Tonellotto. 2022. Lecture notes on neural information retrieval. *arXiv preprint arXiv:2207.13443* (2022).

[57] Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2021. GPL: Generative Pseudo Labeling for Unsupervised Domain Adaptation of Dense Retrieval. *arXiv preprint arXiv:2112.07577* (4 2021). https://arxiv.org/abs/2112.07577

[58] 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).

[59] William Webber, Alistair Moffat, and Justin Zobel. 2010. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems* 28 (2010). https://doi.org/10.1145/1852102.1852106

[60] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. arXiv:2309.07597

[61] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. *arXiv preprint arXiv:2309.03409* (2023).

[62] Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023. Retrieve anything to augment large language models. *arXiv preprint arXiv:2310.07554* (2023).

[63] Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2023. Dense Text Retrieval based on Pretrained Language Models: A Survey. *ACM Transactions on Information Systems* (2023). https://doi.org/10.1145/3637870

[64] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).

[65] Yun Zhou and W. Bruce Croft. 2007. Query performance prediction in web search environments. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '07)*. Association for Computing Machinery, 543–550. https://doi.org/10.1145/1277741.1277835

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

[67] Shengyao Zhuang, Bing Liu, Bevan Koopman, and Guido Zuccon. 2023. Open-source Large Language Models are Strong Zero-shot Query Likelihood Models for Document Ranking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP '23)*. Association for Computational Linguistics, 8807–8817. https://doi.org/10.18653/v1/2023.findings-emnlp.590

[68] Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2023. A Setwise Approach for Effective and Highly Efficient Zero-shot Ranking with Large Language Models. *arXiv preprint arXiv:2310.09497* (2023).

[69] Shengyao Zhuang and Guido Zuccon. 2021. Dealing with Typos for BERT-based Passage Retrieval and Ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP '21)*. Association for Computational Linguistics, 2836–2842.

[70] Shengyao Zhuang and Guido Zuccon. 2022. CharacterBERT and Self-Teaching for Improving the Robustness of Dense Retrievers on Queries with Typos. In *Proceedings of the 45st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. Association for Computing Machinery, 1444–1454.