Route Me If You Can: A Benchmark for Query Reformulation Selection

Hai Son Le
Toronto Metropolitan University

Negar Arabzadeh
UC Berkeley

Amin Bigdeli
University of Waterloo

Radin Hamidi Rad
Mila - Quebec AI Institute

Sajad Ebrahimi
University of Toronto

Charles L. A. Clarke
University of Waterloo

Ebrahim Bagheri
University of Toronto

Abstract

LLM-based query reformulation can improve retrieval effectiveness, but recent work shows that no single reformulation strategy is consistently optimal across queries, domains, retrievers, or model backbones. This creates an inference-time decision problem: *"Given an original query and a pool of candidate reformulations, which one should be issued to the retriever?"* Existing studies are hard to compare because they use different reformulator pools, retrievers, relevance signals, training labels, and evaluation metrics. We introduce QueryRoute, a resource and benchmark for reproducible query reformulation selection. QueryRoute freezes the expensive artifacts needed for this task: original queries, LLM-generated variants, ranked lists under multiple retrievers, retrieval scores, and per-query oracle labels. We revisit query variant selection, a long-studied IR problem, in the setting of LLM-based reformulation, and provide a shared evaluation setup for measuring both retrieval effectiveness and selection behavior. The benchmark spans high-resource web search, heterogeneous zero-shot retrieval, and reasoning-intensive retrieval through TREC DL, BEIR, and BRIGHT-style settings. We instantiate representative baselines from supervised classification, similarity and matrix-factorization routing, QPP, and LLM-as-judge selection. Our empirical characterization shows that query reformulation selection has substantial oracle headroom, indicating significant room for improving over fixed reformulation strategies. At the same time, current selectors recover only part of this potential: selector behavior is strongly conditioned on domain and retriever choice, and mean effectiveness can hide substantially different routing behavior. QueryRoute enables future query-reformulation selectors to be evaluated without regenerating variants, rerunning retrieval, or rebuilding judge pipelines. We have released the codebase, dataset, implementation details, and evaluation scripts at https://github.com/haisonle001/QueryRoute.

1 Introduction

Query reformulation has long been a central component of information retrieval. Classical IR has studied manual, automatic, and interactive query expansion, relevance feedback, pseudo-relevance feedback, term selection, and query refinement as ways to bridge vocabulary mismatch and improve retrieval effectiveness [7, 33, 34]. Selecting among query variants has been studied for decades under headings such as query variation and query-by-query selection [5, 6, 15]. LLM-based reformulation revives this long-standing problem in a new form. Modern pipelines can generate many variants of the same information need using different prompt templates, reformulation objectives, decoding strategies, and model backbones. These variants may include keyword expansions, pseudo-documents, answer-style passages, question decompositions, or corpus-grounded rewrites [17, 27, 36, 37, 43–45]. Recent LLM-based reformulation studies show that no single strategy is consistently optimal across queries, datasets, retrievers, or model backbones [8]. Thus, the question is not only how to generate better reformulations, but also how to choose among available variants for a particular query and retrieval setting.

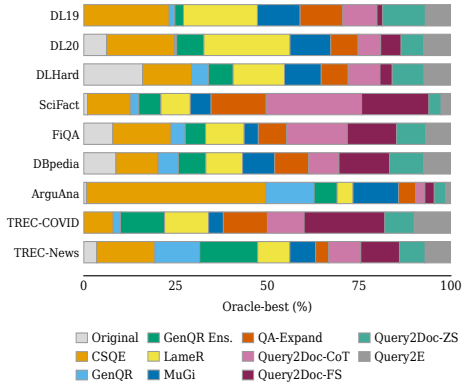

Figure 1: Oracle-winner distribution across different datasets with the `Qwen2.5-7B-Instruct` generator. Different reformulators win on different datasets, highlighting the opportunity for query-reformulation routing.

In this work, we revisit this selection problem under LLM-based query reformulation and study it as query reformulation selection. Given an original query and a fixed pool of candidate variants, a selector must choose one candidate to issue to a retriever, without access to test-set relevance labels. The selected query determines the ranked list returned to the user or passed to downstream components. This task is distinct from evaluating reformulators by average effectiveness. A fixed reformulator can be strong on average while failing on many individual queries; conversely, a selector can improve retrieval by choosing different reformulators for different queries. Figure 1 illustrates this opportunity: oracle-best reformulators are distributed across methods and datasets, showing that different reformulation strategies win in different retrieval settings rather than one strategy dominating throughout. Query reformulation selection is therefore an intra-query decision problem: all candidates express the same underlying information need, but they induce different retrieval outcomes. Several recent works already point toward this decision view. QPP-based variant selection treats

query performance prediction as a mechanism for choosing among multiple variants of the same information need [2]. Learned routing methods, originally developed for selecting among LLMs or systems, can also be adapted to select among reformulators [18, 19, 31]. These methods show that selection matters, but they are difficult to compare because each work typically constructs its own candidate pool, retriever configuration, labels, and evaluation protocol. The missing resource is therefore not another reformulator or another selector, but a benchmark that freezes the expensive intermediate artifacts needed to study selection. This makes comparison expensive and fragile. It also obscures the source of observed gains: improvements may come from the selector, the candidate pool, the retriever, or the evaluation setup.

We present QueryRoute, a resource and benchmark for studying this revived selection problem reproducibly. QueryRoute considers it in the LLM-reformulation setting by freezing the expensive artifacts needed for comparison: original queries, LLM-generated variants, ranked lists under multiple retrievers, retrieval scores, and per-query oracle labels. The benchmark covers 3,757 queries across TREC DL, BEIR, and BRIGHT-style settings; 10 LLM-based reformulators plus the original query; 5 reformulator backbones; and 3 retrievers, yielding 619,905 query–variant–retriever outcomes. Our empirical characterization shows substantial oracle headroom over fixed reformulation strategies, confirming that there is room for improved per-query selection. At the same time, current selectors recover only part of this potential: selector behavior depends strongly on domain and retriever choice, and mean effectiveness can hide substantially different routing behavior. QueryRoute enables future query-reformulation selectors to be evaluated without regenerating variants, rerunning retrieval, or rebuilding judge pipelines. The contributions of this resource are:

- We revisit query variant selection in the context of LLM-based reformulation and treat it as a standardized finite-action benchmark over fixed candidate pools, retrievers, target metrics, and evaluation splits.
- We release a frozen benchmark covering 3,757 queries, 11 candidate systems per query, 5 LLM backbones, and 3 retrievers, for a total of 619,905 retrieval outcomes.
- We provide a standardized evaluation harness that reports both retrieval effectiveness and decision quality, including selection accuracy, near-oracle rate, regret, oracle-gap closure, help/hurt rate, pairwise accuracy, rank correlation, and selection entropy.
- We benchmark selectors from supervised classification, learned routing, QPP, and LLM-as-judge families, showing oracle headroom that no fixed reformulator closes.

2 Related Work

LLM-based query reformulation. LLM reformulators rewrite or enrich queries through keyword generation, pseudo-document generation, multi-question expansion, and corpus-grounded rewriting [17, 27, 36, 37, 43–45]. These methods often improve average retrieval effectiveness, especially for sparse retrieval, but gains vary across queries, datasets, retrievers, and model backbones [8]. This variability motivates per-query selection rather than a single fixed reformulator.

Inference-time selection. A separate line of work decides which reformulation to use at inference time: AdaRewriter ranks rewrites with a trained reward model [25], QueStER learns a query-specification policy for a fixed lexical retriever [35], and ReFormeR applies explicit reformulation patterns via a lightweight per-query selector [9]. A parallel LLM-routing literature selects which model to invoke under cost constraints using matrix-factorization, similarity-weighted, and classifier routers [18, 19, 31]; these transfer naturally to reformulator selection.

Benchmarking query-variant selection. Earlier efforts to capture formulation variance, such as [5], released sets of human-generated variations to evaluate robustness rather than selection. Recently, [2] evaluates QPP as a variant selector for RAG pipelines, releasing variants alongside decision-based metrics. Other resources instead release reformulated queries or query pairs as the primary artifact, optimized toward a fixed objective: QueryGym standardizes LLM-based reformulation methods, prompts, and evaluation [10]; *Matches Made in Heaven* provides supervised query pairs constructed to guarantee effectiveness gains over MS MARCO [1, 30]; and Refairmulate extends this to fairness-aware reformulation [26]. QueryRoute differs by targeting the selection decision, releasing the action–outcome matrix selectors require: variants, ranked lists, retrieval scores, and oracle labels.

Table 1: QueryRoute benchmark. Each query is evaluated across 11 candidate systems, 5 backbone models, and 3 retrievers, yielding 165 query–candidate–backbone–retriever outcomes per query.

Dataset	Queries	Outcomes
TREC DL	147	24,255
BEIR Core	2,861	472,065
BRIGHT	749	123,585
Total	3,757	619,905

3 Query Reformulation Selection

We formulate query reformulation selection as a finite-action decision problem over a fixed pool of query variants. Let q denote an original user query, C a document collection, and R a fixed retriever. A set of reformulation methods

$$\mathcal{E} = \{E_1, \ldots, E_M\}$$

produces a candidate pool

$$\boldsymbol{Q}(q) = \{q_0, q_1, \ldots, q_M\},$$

where $q_0 = q$ is the original query and $q_i = E_i(q)$ for $i > 0$. Each candidate variant induces a ranked list under the retriever:

$$L_i = R(q_i; C) = \langle d_{i,1}, d_{i,2}, \ldots \rangle.$$

For a target metric m, the true utility of candidate q_i is

$$u_i(q; R, m) = m(L_i, \mathrm{qrels}_q).$$

The oracle-best candidate set is then

$$I^{\star}_m(q; R) = \left\{ i \in \{0, \ldots, M\} : u_i(q; R, m) = \max_{j \in \{0,\ldots,M\}} u_j(q; R, m) \right\}.$$

When a single oracle label is needed, we use a deterministic tie-breaking rule, but the evaluation metadata retains all tied oracle-best candidates. A selector S receives the original query, the candidate pool, and any features permitted by the benchmark protocol:

$$\hat{i} = S(q, \boldsymbol{Q}(q), \Phi(q, \boldsymbol{Q}(q), R)),$$

Table 2: Main results for Qwen2.5-7B-Instruct under sparse BM25 retrieval (nDCG@10) across TREC DL, BEIR, and BRIGHT. Best non-oracle per column bold, second-best underlined. Oracle is an upper bound (excluded from ranking).

	TREC DL				BEIR							BRIGHT							
Selector	DL19	DL20	DL-H	DL Avg	SciFact	ArguAna	COVID	FiQA	DBPedia	News	BEIR Avg	Bio	Earth	Econ	Psych	Robot	StackO	Sust	BR Avg
Oracle (upper bound)	0.772	0.750	0.475	0.666	0.801	0.512	0.851	0.364	0.514	0.584	0.604	0.504	0.583	0.334	0.428	0.261	0.348	0.336	0.399
Original query	0.506	0.480	0.285	0.424	0.679	0.397	0.595	0.236	0.318	0.395	0.437	0.182	0.279	0.164	0.134	0.109	0.163	0.161	0.170
Best-single (fixed)	0.687	0.632	0.357	0.559	0.718	**0.434**	0.742	0.246	0.401	0.478	0.503	**0.417**	**0.500**	0.242	**0.363**	0.170	0.249	0.263	**0.315**
Random	0.522	0.497	0.293	0.437	0.698	0.406	0.708	0.230	0.353	0.459	0.476	0.326	0.380	0.186	0.239	0.135	0.215	0.212	0.242
BERT	0.685	0.609	0.332	0.542	0.718	0.401	0.676	0.220	0.375	0.451	0.474	0.377	0.456	0.217	0.275	0.147	0.240	0.242	0.279
SW-Ranking	0.622	0.590	0.328	0.513	0.715	0.406	0.695	0.238	0.369	0.452	0.479	0.405	0.497	0.239	0.340	0.146	0.231	**0.265**	0.303
MF	0.593	0.575	0.327	0.498	0.696	0.396	0.661	0.235	0.363	0.429	0.463	0.308	0.396	0.192	0.222	0.132	0.225	0.201	0.239
Pre-QPP (best)	0.660	0.633	0.328	0.540	0.718	0.432	0.678	0.239	0.400	0.458	0.488	0.399	0.499	0.242	0.327	0.162	0.254	0.263	0.307
Post-QPP (best)	0.679	0.610	0.363	0.551	0.712	0.427	0.659	0.237	0.394	0.454	0.481	0.384	0.492	**0.245**	0.320	0.155	0.237	0.261	0.299
Neural-QPP (best)	**0.717**	0.632	0.362	0.570	0.715	0.423	0.734	0.248	0.389	0.461	0.495	0.376	0.451	0.223	0.301	0.155	0.220	0.237	0.281
LLM-as-judge	0.702	**0.688**	**0.375**	**0.588**	**0.723**	0.417	**0.762**	**0.275**	**0.424**	**0.489**	**0.515**	**0.434**	0.480	0.227	0.361	**0.189**	**0.257**	0.248	0.314

where Φ denotes optional features such as query statistics, retrieval scores, ranked-list properties, learned embeddings, or judge-derived evidence labels. The selected output is the ranked list $L_{\hat{i}}$. The goal of query reformulation selection is to approximate the oracle-best candidate without access to test-set relevance labels:

$$\hat{i}(q) \approx i^{\star}_{m}(q; R).$$

Selection is useful only when variants of the same query induce meaningfully different retrieval outcomes. For an instance (q, C, R, m), the routeability of the candidate pool can be summarized by two gaps: the *oracle headroom* over the original query,

$$\Delta_0(q) = \max_i u_i(q; R, m) - u_0(q; R, m),$$

and the *fixed-policy gap* over the best fixed reformulator,

$$\Delta_{\text{bs}}(q) = \max_i u_i(q; R, m) - u_{\text{bs}}(q; R, m).$$

When $\Delta_{\text{bs}}(q)$ is large, no fixed strategy matches per-query selection; when both gaps are small, a fixed choice already suffices.

4 The Proposed QueryRoute Benchmark

4.1 Purpose and Utility

QueryRoute supports reproducible research on query reformulation selection by fixing the artifacts that are otherwise expensive and inconsistent to recreate—candidate reformulations, ranked lists, retrieval scores, oracle labels, and evaluation splits—so that competing methods are compared over the same action space and retrieval outcomes. This isolates the selector as the object of study and supports selector benchmarking, supervised selector training from per-query oracle labels, and diagnostic analysis of when selection is useful, difficult, or unnecessary.

4.2 Benchmark Construction

QueryRoute materializes the action–outcome matrix used by selectors. For each query, retriever, and target metric, we store the candidate pool, the ranked list induced by each candidate, its retrieval utility, and the oracle-best candidate set. Formally, for each query q we produce $(\mathcal{Q}(q), \{L_i\}, \{u_i\}, I^{\star}_{m}(q))$, where $\mathcal{Q}(q)$ is the candidate pool, L_i is the ranked list produced by candidate i, u_i is its utility, and $I^{\star}_{m}(q)$ is the set of oracle-best candidates under metric m. Construction proceeds in two stages.

Variant Generation. For each query, we construct a pool $\mathcal{Q}(q) = \{q_0, \ldots, q_M\}$ containing the original query q_0 and M LLM-generated reformulations. The reformulations span keyword-level methods (GenQR, GenQR-Ensemble, Query2E), document-level methods (Query2Doc ZS/FS/CoT, QA-Expand, MuGI), and corpus-grounded methods (CSQE, LameR), all generated through QueryGym [10] under a shared prompting and decoding interface. We generate these pools with five backbones—Qwen2.5-7B, Qwen2.5-72B [4], Llama-3.1-8B, Llama-3.3-70B [20], and GPT-4.1 [41]—to support backbone-transfer analysis.Results for all models and retrieval settings are available in our repository.

Retrieval. Each candidate is issued to three first-stage retrievers spanning lexical, learned-sparse, and dense retrieval: BM25, SPLADE, and BGE. Retrieval runs through Pyserini [28] over fixed indexes. For every query–candidate–retriever triple, we store the ranked list and scores, enabling selectors that use ranked-list or score-based features without rerunning retrieval.

Datasets. QueryRoute spans three regimes of increasing difficulty. Web search uses TREC DL19/DL20/DL-Hard over MS MARCO [12, 13, 29, 30] with graded judgments. Zero-shot retrieval uses six BEIR datasets (SciFact, ArguAna, TREC-COVID, FiQA, DBPedia, TREC-News) [23]. Reasoning-intensive retrieval uses seven BRIGHT subsets [39], where relevance requires multi-step reasoning, so retrievers score lower and query reasoning helps disproportionately. Across these, five backbones, ten reformulators, and three retrievers yield the scale in Table 1.

4.3 Baseline Selectors

We include baseline selectors as standard reference points. We include four reference policies: Original, which uses the unreformulated query; Best-single, the dataset-level reformulator with the highest mean nDCG@10 for a given retriever; Random, which samples uniformly; and Oracle, the per-query upper bound. These baselines distinguish gains from reformulation itself from gains due to adaptive selection.

Supervised classifiers. We treat query reformulation selection as a multiclass prediction problem, where each training query is labeled with its oracle-best reformulator under the target metric and retriever. We train a BERT-style classifier from the original query text from MSMARCO [30] and evaluate it in a cross-collection setting, testing whether selector behavior transfers beyond the source-domain oracle labels.

Similarity and matrix-factorization routers. We adapt model-routing methods to reformulator selection by treating each reformulator as an expert. The similarity-weighted router embeds queries with BGE [11] and selects reformulators that performed well on nearby training queries. The matrix-factorization router learns latent query and reformulator representations from oracle-derived preferences. Both methods select from query-side signals and do not inspect retrieved documents at test time.

Query performance prediction selectors. QPP selectors estimate the expected effectiveness of each candidate variant and

Table 3: Cross-retriever and cross-regime transfer for Qwen2.5-7B-Instruct (nDCG@10, averaged over datasets within each regime). Best non-oracle per column bold, second-best underlined.

	TREC DL (avg)			BEIR (avg)		
Selector family	BM25	BGE	SPLADE	BM25	BGE	SPLADE
Oracle (upper bound)	0.666	0.689	0.722	0.604	0.671	0.587
Original query	0.423	0.586	**0.612**	0.437	0.569	**0.525**
Best-single (fixed)	0.559	**0.600**	**0.612**	**0.503**	**0.585**	**0.525**
Supervised	0.542	0.572	0.483	0.473	0.575	0.280
SW-Ranking	0.514	0.376	0.517	0.479	0.565	0.220
MF	0.498	0.418	0.523	0.463	0.563	0.271
Pre-QPP (best)	0.540	0.017	0.531	0.488	0.572	0.240
Post-QPP (best)	0.551	0.350	0.517	0.480	0.577	0.257
Neural-QPP (best)	0.570	0.269	0.525	0.495	0.574	0.249
LLM-as-judge	**0.588**	0.396	0.535	**0.515**	0.571	0.247

choose the highest-scoring one. We include pre-retrieval predictors based on query and collection statistics, post-retrieval predictors based on ranked-list and score-distribution signals, and neural predictors based on BERT-QPP-style query–document interactions [3, 14, 22, 24, 32, 38, 40, 46, 47]. For compact reporting, Pre-QPP (best), Post-QPP (best), and Neural-QPP (best) denote the strongest predictor identified within each QPP family after evaluation; these rows are diagnostic within-family upper bounds rather than deployable selector choices.

Outcome-level judge selectors. Outcome-level judge selectors estimate the utility of the ranked lists induced by each candidate. We use UMBRELA-style graded relevance labels in $\{0, 1, 2, 3\}$ to score retrieved documents and select the candidate with the highest estimated ranking utility [21, 42]. Documents are judged against the original query rather than each reformulated variant, allowing shared documents across candidate lists to be judged once. Because this family relies on additional relevance estimation, we report it separately when discussing compute requirements. We adopt `DeepSeek-V3` [16] as the default judge following [21].

4.4 Evaluation Protocol

We report two metric classes. **Retrieval effectiveness** uses nDCG@10 as primary, with MAP@1000 and Recall@1000 as extended metrics. **Decision quality** uses three compact diagnostics: Near-Or (selected candidate within $\varepsilon = 0.05$ nDCG@10 of the per-query oracle) and Help/Hurt (fraction better/worse than the original query). Together, these show whether a selector approaches the oracle while improving over a strong fixed reformulator. Additional diagnostics are provided in our repository.

5 Benchmarking the QueryRoute dataset

We use the released QueryRoute artifacts to characterize how much headroom exists for query reformulation selection and how much of that headroom current selector families recover. Table 2 reports the main BM25 results using `Qwen2.5-7B-Instruct`. **First**, across all three regimes the pool shows substantial oracle headroom: on TREC DL, Original/Best-single/Oracle reach 0.424/0.559/0.666, with similar gaps on BEIR and BRIGHT. No fixed reformulator closes the oracle gap. LLM-as-judge is the strongest non-oracle selector on TREC DL (0.588) and BEIR (0.515); on BRIGHT it ties Best-single (0.314 vs. 0.315). QueryRoute is thus clearly routeable, yet current selectors recover only part of the available headroom. **Second**, Table 3 shows that selector behavior is strongly

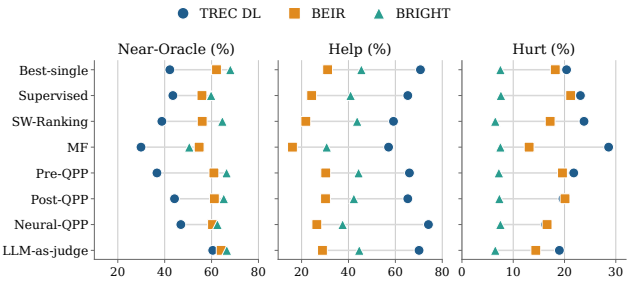

Figure 2: Decision quality on TREC DL/ BEIR/ BRIGHT. Averages are query-weighted within each regime. Hurt is lower better; higher is better for the other metrics.

conditioned on the retriever, and that the advantage of learned selection over fixed baselines does not transfer beyond sparse retrieval. Under BM25, LLM-as-judge is the best non-oracle method on both TREC DL (0.588 nDCG@10) and BEIR (0.515), exceeding Best-single. Under BGE and SPLADE, however, no learned selector beats the fixed baselines: on TREC DL, Best-single reaches 0.600 under BGE while Original and Best-single tie at 0.612 under SPLADE; on BEIR, Best-single leads under BGE (0.585) and ties Original under SPLADE (0.525). The effect is most severe for BEIR/SPLADE, where learned selectors collapse to 0.22–0.28 against a 0.525 baseline. These results indicate that query reformulation selection cannot be evaluated in a single sparse-retrieval setting: a selector that appears effective under BM25 may fail to transfer to dense or learned-sparse retrieval, and stronger retrievers can erase the benefit of adaptive reformulation altogether. **Finally**, Figure 2 shows that retrieval effectiveness alone does not fully characterize selector behavior. On TREC DL, LLM-as-Judge attains the highest *near-oracle* rate of 60.5%, but Neural-QPP provides a better help/hurt profile, with 74.1% help and 16.3% hurt compared with 70.1% and 19.0% for LLM-as-Judge. Similar trends appear on BEIR and BRIGHT, where selectors with comparable mean nDCG@10 exhibit different near-oracle, help, and hurt rates. For example, on BRIGHT, LLM-as-Judge and Best-Single achieve similar near-oracle rates (66.5% vs. 68.0%) with nearly identical hurt rates. These results show that mean effectiveness can mask query-level routing differences, motivating decision-quality diagnostics beyond aggregate retrieval metrics.

6 Conclusion

We presented QueryRoute, a resource for reproducible query reformulation selection that freezes the expensive action–outcome matrix across web search, zero-shot, and reasoning-intensive regimes, turning selector evaluation into a read-only decision problem. Benchmarking representative selector families shows substantial oracle headroom over both the original query and the best fixed reformulator, but also shows that current selectors recover only part of this potential. Selector-family rankings shift across retrievers and regimes, and mean effectiveness can hide different routing behavior, motivating decision-quality metrics in addition to retrieval effectiveness. QueryRoute therefore provides a shared substrate for comparing future selectors while making clear that its oracle is bounded by the released candidate pool. Expanding reformulator diversity, retriever coverage, backbone coverage, and multilingual evaluation are important directions for future releases.

GenAI Usage Disclosure

Large language models were used for light editing of author-written text, including grammar correction, improving fluency. All text reflects the authors' own ideas; no sections were produced entirely by a generative model. GenAI tools were also used to assist with coding tasks; all code was reviewed and validated by the authors

References

[1] Negar Arabzadeh, Amin Bigdeli, Shirin Seyedsalehi, Morteza Zihayat, and Ebrahim Bagheri. 2021. Matches Made in Heaven: Toolkit and Large-Scale Datasets for Supervised Query Reformulation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4417–4425. doi:10.1145/3459637.3482009

[2] Negar Arabzadeh, Andrew Drozdov, Michael Bendersky, and Matei Zaharia. 2026. Can QPP Choose the Right Query Variant? Evaluating Query Variant Selection for RAG Pipelines. arXiv:2604.22661 [cs.IR] https://arxiv.org/abs/2604.22661

[3] Negar Arabzadeh, Maryam Khodabakhsh, and Ebrahim Bagheri. 2021. BERT-QPP: Contextualized Pre-trained Transformers for Query Performance Prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2857–2861. doi:10.1145/3459637.3482063

[4] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen Technical Report. arXiv:2309.16609 [cs.CL] https://arxiv.org/abs/2309.16609

[5] Peter Bailey, Alistair Moffat, Falk Scholer, and Paul Thomas. 2016. UQV100: A Test Collection with Query Variability. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Pisa, Italy) *(SIGIR '16)*. Association for Computing Machinery, New York, NY, USA, 725–728. doi:10.1145/2911451.2914671

[6] Rodger Benham, Joel Mackenzie, Alistair Moffat, and J. Shane Culpepper. 2019. Boosting Search Performance Using Query Variations. *ACM Transactions on Information Systems* 37, 4 (Oct. 2019), 1–25. doi:10.1145/3345001

[7] Jagdev Bhogal, Andrew MacFarlane, and Peter Smith. 2007. A review of ontology based query expansion. *Information processing & management* 43, 4 (2007), 866–886.

[8] Amin Bigdeli, Radin Hamidi Rad, Hai Son Le, Mert Incesu, Negar Arabzadeh, Charles LA Clarke, and Ebrahim Bagheri. 2026. A Reproducibility Study of LLM-Based Query Reformulation. *arXiv preprint arXiv:2604.27421* (2026).

[9] Amin Bigdeli, Mert Incesu, Negar Arabzadeh, Charles L. A. Clarke, and Ebrahim Bagheri. 2026. *ReFormeR: Learning and Applying Explicit Query Reformulation Patterns*. Springer Nature Switzerland, 400–408. doi:10.1007/978-3-032-21300-6_30

[10] Amin Bigdeli, Radin Hamidi Rad, Mert Incesu, Negar Arabzadeh, Charles L. A. Clarke, and Ebrahim Bagheri. 2025. QueryGym: A Toolkit for Reproducible LLM-Based Query Reformulation. arXiv:2511.15996 [cs.IR] https://arxiv.org/abs/2511.15996

[11] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2025. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. arXiv:2402.03216 [cs.CL] https://arxiv.org/abs/2402.03216

[12] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. *CoRR* abs/2102.07662 (2021). arXiv:2102.07662 https://arxiv.org/abs/2102.07662

[13] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the TREC 2019 deep learning track. *arXiv preprint arXiv:2003.07820* (2020).

[14] 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*. 299–306.

[15] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. 2004. A framework for selective query expansion. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management* (Washington, D.C., USA) *(CIKM '04)*. Association for Computing Machinery, New York, NY, USA, 236–237. doi:10.1145/1031171.1031220

[16] DeepSeek-AI and Aixin Liu et al. 2025. DeepSeek-V3 Technical Report. arXiv:2412.19437 [cs.CL] https://arxiv.org/abs/2412.19437

[17] Kaustubh D Dhole and Eugene Agichtein. 2024. Genqrensemble: Zero-shot llm ensemble prompting for generative query reformulation. In *European Conference on Information Retrieval*. Springer, 326–335.

[18] Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. 2024. Hybrid LLM: Cost-Efficient and Quality-Aware Query Routing. arXiv:2404.14618 [cs.LG] https://arxiv.org/abs/2404.14618

[19] Dujian Ding, Ankur Mallick, Shaokun Zhang, Chi Wang, Daniel Madrigal, Mirian Del Carmen Hipolito Garcia, Menglin Xia, Laks V. S. Lakshmanan, Qingyun Wu, and Victor Rühle. 2025. BEST-Route: Adaptive LLM Routing with Test-Time Optimal Compute. arXiv:2506.22716 [cs.LG] https://arxiv.org/abs/2506.22716

[20] Aaron Grattafiori et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] https://arxiv.org/abs/2407.21783

[21] Naghmeh Farzi and Laura Dietz. 2025. Does UMBRELA work on other LLMs?. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Padua, Italy) *(SIGIR '25)*. Association for Computing Machinery, New York, NY, USA, 3214–3222. doi:10.1145/3726302.3730317

[22] Ben He and Iadh Ounis. 2004. Inferring Query Performance Using Pre-retrieval Predictors. In *String Processing and Information Retrieval, 11th International Conference, SPIRE 2004*. 43–54. doi:10.1007/978-3-540-30213-1_5

[23] Ehsan Kamalloo, Nandan Thakur, Carlos Lassance, Xueguang Ma, Jheng-Hong Yang, and Jimmy Lin. 2024. Resources for Brewing BEIR: Reproducible Reference Models and Statistical Analyses. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 1431–1440. doi:10.1145/3626772.3657862

[24] Kuilam L. Kwok. 1996. A New Method of Weighting Query Terms for Ad-hoc Retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 187–195.

[25] Yilong Lai, Jialong Wu, Zhenglin Wang, and Deyu Zhou. 2025. AdaRewriter: Unleashing the Power of Prompting-based Conversational Query Reformulation via Test-Time Adaptation. arXiv:2506.01381 [cs.CL] https://arxiv.org/abs/2506.01381

[26] Hai Son Le, Shirin Seyedsalehi, Morteza Zihayat, and Ebrahim Bagheri. 2026. Refairmulate: A Large-Scale Dataset for Gender-Fair Query Reformulations. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*. To appear.

[27] Yibin Lei, Yu Cao, Tianyi Zhou, Tao Shen, and Andrew Yates. 2024. Corpus-Steered Query Expansion with Large Language Models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, Yvette Graham and Matthew Purver (Eds.). Association for Computational Linguistics, St. Julian's, Malta, 393–401. doi:10.18653/v1/2024.eacl-short.34

[28] 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* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 2356–2362. doi:10.1145/3404835.3463238

[29] Iain Mackie, Jeffrey Dalton, and Andrew Yates. 2021. How deep is your learning: The DL-HARD annotated deep learning dataset. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2335–2341.

[30] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human-generated machine reading comprehension dataset. (2016).

[31] Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. 2025. RouteLLM: Learning to Route LLMs with Preference Data. arXiv:2406.18665 [cs.LG] https://arxiv.org/abs/2406.18665

[32] Jay M. Ponte and W. Bruce Croft. 2017. A Language Modeling Approach to Information Retrieval. *ACM SIGIR Forum* 51 (2017), 202–208.

[33] Yonggang Qiu and Hans-Peter Frei. 1993. Concept based query expansion. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*. 160–169.

[34] Joseph John Rocchio Jr. 1971. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing* (1971).

[35] Arthur Satouf, Yuxuan Zong, Habiboulaye Amadou Boubacar, Pablo Piantanida, and Benjamin Piwowarski. 2026. QueStER: Query Specification for Generative Keyword-Based Retrieval. In *Findings of the Association for Computational Linguistics: EACL 2026*, Vera Demberg, Kentaro Inui, and Lluís Marquez (Eds.). Association for Computational Linguistics, Rabat, Morocco, 5957–5968. doi:10.18653/v1/2026.findings-eacl.312

[36] Wonduk Seo and Seunghyun Lee. 2025. QA-Expand: Multi-Question Answer Generation for Enhanced Query Expansion in Information Retrieval. *arXiv preprint arXiv:2502.08557* (2025).

[37] Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Yibin Lei, Tianyi Zhou, Michael Blumenstein, and Daxin Jiang. 2024. Retrieval-Augmented Retrieval: Large Language Models are Strong Zero-Shot Retriever. In *Findings of the Association for Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok,

Thailand, 15933–15946. doi:10.18653/v1/2024.findings-acl.943

[38] 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, 2 (2012), 1–35.

[39] Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han yu Wang, Haisu Liu, Quan Shi, Zachary S. Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan O. Arik, Danqi Chen, and Tao Yu. 2025. BRIGHT: A Realistic and Challenging Benchmark for Reasoning-Intensive Retrieval. arXiv:2407.12883 [cs.CL] https://arxiv.org/abs/2407.12883

[40] 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*. 1891–1894.

[41] OpenAI team. 2024. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL] https://arxiv.org/abs/2303.08774

[42] Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Nick Craswell, and Jimmy Lin. 2024. UMBRELA: UMbrela is the (Open-Source Reproduction of the) Bing RELevance Assessor. *arXiv:2406.06519* (2024).

[43] Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. *arXiv preprint arXiv:2303.07678* (2023).

[44] Xiao Wang, Sean MacAvaney, Craig Macdonald, and Iadh Ounis. 2023. Generative query reformulation for effective adhoc search. *arXiv preprint arXiv:2308.00415* (2023).

[45] Le Zhang, Yihong Wu, Qian Yang, and Jian-Yun Nie. 2024. Exploring the Best Practices of Query Expansion with Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 1872–1883. doi:10.18653/v1/2024.findings-emnlp.103

[46] Ying Zhao, Falk Scholer, and Yohannes Tsegay. 2008. Effective Pre-retrieval Query Performance Prediction Using Similarity and Variability Evidence. In *Advances in Information Retrieval: 30th European Conference on IR Research, ECIR 2008*. 52–64. doi:10.1007/978-3-540-78646-7_8

[47] 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*. 543–550.