EviQE: Evidence Selection for LLM-Based Query Expansion

Abstract

Recent query expansion methods increasingly rely on Large Language Models (LLMs) to generate reformulations conditioned on documents retrieved from the target corpus. These documents are provided to the language model as context and constitute the information available during reformulation. While prior work has explored diverse reformulation strategies and iterative retrieval-generation pipelines, comparatively little attention has been paid to the selection of the documents used to condition generation. Yet the quality of a reformulation depends not only on the language model itself, but also on the documents it receives as input. In this paper, we investigate whether improvements in query expansion can be achieved through better document selection rather than more sophisticated reformulation strategies. We propose EviQE, a query expansion framework that aggregates documents retrieved by multiple reformulation methods, selects a compact set of informative passages, and provides them to the language model to guide a single grounded reformulation step. This formulation treats reformulators as complementary retrieval perspectives rather than alternative query generators and explicitly separates document selection from reformulation generation. Experiments on three TREC DL collections and five BEIR benchmarks show that different reformulation methods frequently retrieve complementary relevant documents, allowing pooled candidate sets to achieve substantially higher relevant-document coverage than any individual reformulator. Across all benchmarks, EviQE consistently outperforms direct reformulation, cold-start expansion, and single-source seeded expansion methods. We further show that once high-quality conditioning documents have been identified, additional retrieval-generation iterations provide limited benefit and often reduce effectiveness.

1 Introduction

Query reformulation has been central to improving information retrieval [2, 21, 22] through approaches such as reducing vocabulary mismatch, exposing latent aspects of the query, or aligning queries with the target corpus. LLMs have expanded the space of reformulation strategies by generating pseudo-documents, hypothetical answers, reasoning traces, and corpus-aware rewrites to be used as retrieval surrogates or expansion text [8, 13, 23, 29]. Methods such as Query2Doc [28], HyDE [12], CSQE [16], LameR [24], MuGI [30], and ThinkQE [17] show LLM-generated expansions improve first-stage retrieval without modifying the retrieval method.

Existing expansion methods differ in prompting, outputs, and retrieval-generation rounds, but largely focus on how reformulations are generated. In corpus-aware approaches such as CSQE and LameR, generation is conditioned on documents retrieved from the target corpus [12, 16, 24, 28], so the reformulation depends on both the language model and the documents supplied as context. These documents constitute the **evidence** available during generation, yet comparatively little attention has been paid to how these documents are selected. Since different retrieval strategies surface different documents for the same query, expansion effectiveness may depend not only on the generated text but also on the documents used to generate it.

Most corpus-aware expansion methods condition generation on a single set of retrieved documents obtained from the original query, a particular reformulation strategy, or a previous iteration of an expansion pipeline. Existing approaches also commonly treat reformulators as competing alternatives. AdaRewriter and ReFormeR [3, 15] explicitly select one strategy per query, even though different reformulators often retrieve different relevant documents for the same need. Iterative methods such as ThinkQE [17] repeatedly retrieve, read, and reformulate, but each round simultaneously changes the expansion and adds retrieved documents, so observed gains may stem from successive reformulations, additional documents, or both, raising whether similar gains are achievable through stronger document selection before generation. We are therefore motivated to view query expansion as a document-selection problem, rather than a pure generation one.

Motivated by this perspective, we propose EviQE, a query expansion framework that explicitly separates document selection from expansion generation. In EviQE, multiple reformulators are executed independently, the documents they retrieve are aggregated into a unified candidate pool, and a compact set of informative passages is selected prior to a single grounded generation step. Unlike reformulator-selection approaches that choose a single rewriting strategy or rank-fusion approaches that combine retrieval results after ranking, EviQE selects the retrieved documents that will be provided to the LLM before expansion generation takes place. Rather than viewing reformulators solely as alternative query generators, EviQE treats them as complementary retrieval perspectives that expose different documents relevant to the same information need. The key contributions of this work are summarized as follows.

- We reformulate LLM-based query expansion as a document-selection problem and provide a framework for disentangling the contribution of conditioning documents from that of reformulation generation.
- We propose EviQE, a document-centric query expansion framework that aggregates documents retrieved by multiple reformulation methods and selects a compact set of passages prior to grounded generation.
- We provide an empirical analysis of reformulator complementarity and show that different reformulation methods frequently expose distinct relevant documents for the same information need.
- We evaluate EviQE on TREC DL and BEIR benchmarks and show that aggregating and selecting retrieved documents from multiple reformulators consistently improves retrieval effectiveness while reducing the benefit of additional retrieval-generation iterations.

Our results show that different reformulation methods frequently retrieve complementary relevant documents, allowing pooled candidate sets to achieve substantially higher relevant-document coverage than any individual reformulator. Selecting passages from these pools consistently improves retrieval effectiveness relative to cold-start expansion, direct reformulation, and single-source seeded expansion. Across the evaluated benchmarks, our approach yields

Table 1: Reformulator complementarity analysis. QCov and DocCov measure query- and document-level coverage, respectively; Δ_{pool} denotes the recall gain of pooled retrieval over the best individual reformulator.

Dataset	QCov	DocCov	Jaccard	$\lvert D(q)\rvert$	BestSingle	PoolRecall	Δ_{pool}	Density
TREC DL 2019	0.964	0.072	0.342	36.6	0.172	0.325	+0.153	0.557
TREC DL 2020	0.951	0.092	0.406	32.7	0.223	0.347	+0.124	0.486
DL-Hard	0.693	0.088	0.329	38.7	0.180	0.285	+0.105	0.229
SciFact	0.858	0.826	0.552	22.6	0.863	0.921	+0.058	0.050
TREC-COVID	0.991	0.015	0.257	41.9	0.021	0.064	+0.043	0.675
FiQA	0.467	0.236	0.379	32.4	0.308	0.445	+0.137	0.035
DBPedia-Entity	0.861	0.079	0.354	35.7	0.244	0.393	+0.149	0.213
TREC-News	0.944	0.101	0.477	27.1	0.172	0.281	+0.109	0.436

Table 2: Main retrieval results (nDCG@10). Cold denotes ThinkQE seeded from the initial BM25 ranking. Bold indicates the best non-oracle result. Average rows are reported separately for TREC DL and BEIR benchmarks. Improvements of LLM-Score over Best Single and Best QR are significant on both average rows (paired per-query t-test $p < 0.05$).

	Direct / cold		*Single-source seeded*		*Portfolio-selected documents*		
Dataset	Cold	Best QR	Single mean	Best single	Freq	RRF	LLM-Score
TREC DL 2019	0.6673	0.6798	0.6573	0.6714	0.6874	0.6895	**0.6961**
TREC DL 2020	0.6321	0.6322	0.6303	0.6384	0.6313	0.6368	**0.6603**
DL-Hard	0.3489	0.3470	0.3497	0.3614	0.3659	0.3597	**0.3718**
SciFact	0.7376	0.7183	0.7373	0.7400	0.7352	0.7363	**0.7435**
TREC-COVID	0.7494	0.7423	0.7515	0.7711	0.7572	0.7511	**0.7760**
FiQA	0.2575	0.2460	0.2560	0.2585	0.2536	0.2568	**0.2785**
DBPedia-Entity	0.4264	0.3987	0.4283	0.4314	0.4287	0.4273	**0.4410**
TREC-News	0.5004	0.4778	0.4990	0.5082	0.4901	0.4924	**0.5174**
DL Avg.	0.5494	0.5530	0.5458	0.5571	0.5615	0.5620	**0.5760**
BEIR Avg.	0.5343	0.5166	0.5344	0.5419	0.5330	0.5328	**0.5513**

the strongest and most consistent improvements, while additional retrieval rounds provide limited performance improvements once strong conditioning documents have been identified.

2 Methodology

EviQE is motivated by the observation that different reformulation methods retrieve different relevant documents for the same need, yet existing query expansion methods condition generation on a single retrieved set. EviQE formulates LLM-based query expansion as an inference-time document-selection problem. Given a query q, corpus C, fixed retriever R, and fixed generator G, let $S(q)$ denote the set of retrieved documents supplied to the generator. These retrieved documents constitute the evidence available to the language model during expansion generation. The generator produces an expansion $e = G(q, S(q))$, which is appended to the original query to form the expanded query $\hat{q} = q \oplus e$. The expanded query is then submitted to the retriever, yielding the final ranking $L(\hat{q}) = R(\hat{q}; C)$. The objective is to identify the set of retrieved documents that, when provided to the generator, produces the most effective expanded query.

$$S^\star(q) = \arg\max_{S \in \Omega_B(q)} \mathcal{M}\big(R(q \oplus G(q,S); C), y^\star(q)\big), \tag{1}$$

where $\Omega_B(q)$ denotes the family of candidate subsets drawn from the retrieved-document pool, $\mathcal{M}$ is the retrieval metric, and $y^\star(q)$ denotes the unknown relevance labels associated with query q.

Since $y^\star(q)$ is unavailable at inference time, EviQE approximates $S^\star(q)$ by constructing a candidate pool from multiple reformulator probes and selecting a compact subset before generation. This separates two decisions that are usually coupled in LLM-based expansion, namely what the generator reads and how it rewrites. We keep the generator fixed and vary only the selected documents.

2.1 The EviQE Framework

To approximate the optimal document set, EviQE constructs a candidate pool of retrieved documents from multiple reformulation strategies and selects a compact subset to condition generation.

Let $\mathcal{P} = \{m_0, m_1, \ldots, m_M\}$ denote a portfolio of reformulators, where $m_0(q) = q$ corresponds to the original query. Each reformulator produces a query variant $q_i = m_i(q)$, which is independently submitted to the retriever to obtain a top-K ranking,

$$L_i(q) = \text{TopK}(R(q_i; C)). \tag{2}$$

Rather than selecting a single reformulation strategy, EviQE retains the documents retrieved by all reformulators. The resulting candidate pool is defined as

$$D(q) = \bigcup_{i=0}^{M} L_i(q). \tag{3}$$

The union operation produces a de-duplicated set of candidate documents while preserving source-specific rank information. Unlike reformulator-selection approaches that retain only a single reformulation strategy [3, 15], EviQE preserves the documents surfaced by all reformulators in the portfolio. Since different reformulators frequently retrieve different relevant documents, the resulting candidate pool often provides broader coverage of the information need than any individual reformulator.

Because the candidate pool is typically larger than the amount of information that can be supplied to the language model, EviQE subsequently performs document selection. Given a selector A with scoring function $a_A(q, d)$, the framework selects the top-B documents according to:

$$S_A(q) = \underset{d \in D(q)}{\text{Top-B}}\, a_A(q, d), \tag{4}$$

where B denotes the document budget. All candidate documents are scored with respect to the original query rather than the reformulated query that retrieved them. This design anchors document selection to the user's information need and reduces sensitivity to reformulation drift. The selected document set $S_A(q)$ is then supplied to a fixed generator, which produces an expansion:

$$e = G(q, S_A(q)). \tag{5}$$

The generated expansion is appended to the original query and submitted to the retriever for final ranking. Because the retriever, generator, prompt, and document budget remain fixed across selectors, differences in retrieval effectiveness can be attributed directly to the quality of the selected documents.

2.2 Evidence Selection Strategies

Since the effectiveness of EviQE depends on the quality of the selected evidence, we instantiate a_A with three evidence-selection strategies that prioritize different signals of document utility: agreement, rank position, and estimated relevance. These let us examine whether useful expansion evidence is primarily characterized by repeated discovery across reformulators, strong retrieval rankings, or semantic relevance to the original query.

Agreement-based Selection. Documents are scored by cross-source agreement, assigning higher scores to those retrieved by more reformulators, under the assumption that documents repeatedly retrieved across reformulation perspectives are more likely

Table 3: Candidate pool ablation.

#	Candidate pool	DL19	DL20	DL-H	DL Avg.
1	BM25@100 + LLM-score	0.664	0.618	0.337	0.540
4	Original, CSQE, LameR, MuGI	0.693	0.658	0.377	0.576
6	+ Query2Doc (CoT, ZS)	0.705	0.665	0.372	**0.581**
8	+ QA-expand, Query2Doc (FS)	0.698	0.657	0.376	0.577
11	+ GenQR, GenQR Ens., Query2E	0.696	0.660	0.372	0.576

#	Candidate pool	SciFact	COVID	FiQA	DBPed	News	BEIR Avg.
1	BM25@100 + LLM-score	0.747	0.751	0.266	0.417	0.495	0.535
4	Original, CSQE, LameR, MuGI	0.749	0.760	0.271	0.438	0.509	0.545
6	+ Query2Doc (CoT, ZS)	0.741	0.757	0.273	0.440	0.511	0.545
8	+ QA-expand, Query2Doc (FS)	0.746	0.765	0.275	0.441	0.519	0.549
11	+ GenQR, GenQR Ens., Query2E	0.744	0.776	0.279	0.441	0.517	**0.551**

broadly relevant to the information need. Ties are broken by the best rank assigned by any reformulator.

Rank-based Selection. This strategy uses Reciprocal Rank Fusion (RRF) [4] to combine cross-source agreement with rank position, so documents retrieved by multiple reformulators and ranked near the top score highest. The rationale is that rank provides a utility signal beyond agreement alone, prioritizing highly ranked evidence even when reformulator overlap is limited.

Relevance-based Selection. This strategy directly estimates relevance using an LLM judge that assigns a graded relevance score [10, 18, 20, 25–27] to each pooled document with respect to the original query. Unlike the previous strategies, which use retrieval behavior as a proxy for relevance, it explicitly evaluates the semantic relationship between document and information need. Since the pooled set is de-duplicated, each unique document is evaluated only once.

3 Experimental Setup

Datasets and evaluation. We use TREC DL 2019, DL 2020, DL-Hard [5, 6, 19] and five BEIR collections [14]. Following prior reformulation work [13, 16, 24, 28, 30], nDCG@10 is adopted as the primary metric; other metrics reported on our repository.

Large Language Models. Our framework uses LLMs in two roles, namely as reformulators/generators producing query expansions, and as judges estimating document relevance. Unless noted, all reformulation and expansion experiments use `Qwen2.5-7B-Instruct` [1]. We additionally use `DeepSeek-V3` [7] and `Llama-3.3-70B` [9] as generators, with the judge held fixed as robustness check. For document scoring, we use the UMBRELA relevance-judging framework [27]; following recent reproducibility findings [11], we adopt `DeepSeek-V3` as the default judge for its strong agreement with proprietary assessors on TREC DL.

Implementation details. All methods use the same BM25 retriever, document index, generator, expansion prompt, and final retrieval pipeline. Each reformulated query retrieves the top $K = 10$ passages, and the selector chooses the top $B = 10$ passages for generation. The portfolio contains the identity query and ten reformulation sources, namely CSQE, LameR, MuGI, Query2Doc-ZS, Query2Doc-FS, Query2Doc-CoT, QA-Expand, GenQR, GenQR-Ensemble, and Query2E. Each generator call produces two expansion candidates, appended to the original query using the same formatting for all methods [17]. Compared with a T-round ThinkQE pipeline, EviQE uses one generation round, $|\mathcal{P}|$ parallel retrievals, and up to $|D(q)|$ judge calls. For the judge, we employ graded relevance assessments ($y \in \{0, 1, 2, 3\}$). All experimental results are averaged over three independent runs. Code, runs, prompts, and detailed results are released anonymously[1].

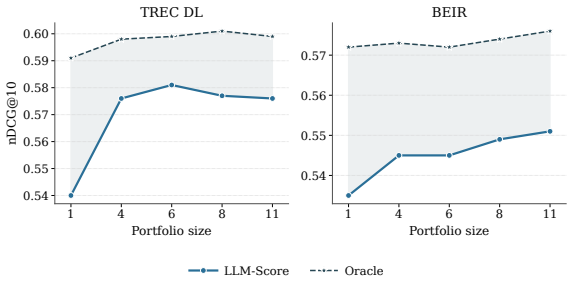

Figure 1: Effect of portfolio size on retrieval effectiveness (nDCG@10), averaged over TREC DL (left) and BEIR (right).

Baselines. We compare our work against three families of methods. (i) **Direct reformulation**: each of the ten reformulators is issued directly to BM25 without iterative expansion; Best QR reports the strongest reformulator per dataset. (ii) **Iterative expansion**: ThinkQE, a standard iterative baseline, initialized from the BM25 ranking of the original query without portfolio evidence. (iii) **Single-source seeded expansion**: each reformulator independently seeds the same ThinkQE-style generator; we report Single-mean (average across sources) and Best single (best source per dataset).

4 Results and Findings

Our experiments are organized around 3 main research questions: **RQ1.** Do different reformulation methods retrieve the same relevant documents, or do they expose complementary documents for the same information need? **RQ2.** Does selecting documents from a reformulator portfolio improve query expansion effectiveness, and are any gains attributable to document selection rather than a stronger reformulator or a larger candidate pool? **RQ3.** Once high-quality conditioning documents have been identified, do additional retrieval-generation rounds provide further benefit?

RQ1: Do reformulators retrieve complementary documents?

RQ1 examines the degree of complementarity among the documents retrieved by different reformulation methods, asking whether reformulators agree at the query level while diverging at the document level. If they improve largely the same queries but surface different passages, selecting a single reformulator discards useful information; if they retrieve the same documents, aggregation offers little benefit. We make three observations.

First, reformulators frequently converge at the query level while diverging at the document level. As shown in Table 1, query hit rates are consistently high (0.964 on TREC DL 2019, 0.951 on DL 2020, 0.991 on TREC-COVID, 0.944 on TREC-News), yet per-source document coverage is far lower (0.072, 0.092, 0.015, 0.101). Their complementarity thus arises primarily from the documents they retrieve rather than the queries they address. **Second**, pooling documents across reformulators consistently exposes more relevant material than any single strategy, improving on the strongest individual source on every benchmark, with absolute recall gains from +0.043 (TREC-COVID) to +0.153 (DL 2019); BEIR shows similar trends (+0.137 on FiQA, +0.149 on DBPedia-Entity). **Third**, the

[1] https://github.com/queryreform-judge/EviQE

Table 4: Model robustness on TREC DL under `DeepSeek-V3` and `Llama-3.3-70B` as generator.

Model	Strategy	DL19	DL20	DL-H	DL Avg.
`DeepSeek-V3`	Best QR	**0.709**	0.648	0.364	0.574
	ThinkQE	0.662	0.641	0.362	0.555
	Frequency	0.679	0.638	0.368	0.562
	RRF	0.675	0.638	0.356	0.556
	LLM-Score	0.700	0.651	**0.373**	**0.575**
`Llama-3.3-70B`	Best QR	0.671	**0.663**	0.367	0.567
	ThinkQE	0.674	0.630	0.370	0.558
	Frequency	0.681	0.638	0.369	0.563
	RRF	0.683	0.632	0.372	0.562
	LLM-Score	**0.688**	0.651	0.370	**0.570**

pooled set is not uniformly informative, i.e., the de-duplicated pool averages 22.6–41.9 passages per query with considerable density variation (dense on TREC-COVID, sparse on FiQA and SciFact). Because the LLM can consume only a limited number of passages, aggregation alone is insufficient and expansion effectiveness also depends on identifying the most informative passages from the pool, not merely their quantity.

RQ2: Where do document-selection gains come from?

We compare EviQE against increasingly strong baselines that isolate whether gains arise from a stronger reformulated query, a stronger individual source, pooling alone, or the combination of pooling and selection in EviQE. Table 2 reports results across all eight benchmarks, supporting the following observations.

First, portfolio-based selection consistently improves one-shot expansion. Relative to ThinkQE, EviQE with relevance-based selection raises the DL average from 0.5494 to 0.5760 (+4.8%) and the BEIR average from 0.5343 to 0.5513 (+3.2%), with gains on all three DL and all five BEIR benchmarks, despite a single expansion round and unchanged retriever, generator, prompt, and document budget. **Second**, the gains are not explained by a better reformulated query alone. The Best QR baseline, controlling for the strongest reformulation issued directly to BM25, trails EviQE on both DL (0.5530 vs. 0.5760) and BEIR (0.5166 vs. 0.5513). The benefit comes from using reformulators as complementary retrieval perspectives whose documents guide expansion. **Third**, the gains are not from selecting a strong individual source. The Single-mean baseline ties Cold on BEIR (0.5344 vs. 0.5343), and even Best Single, assuming oracle access to the strongest source, remains below EviQE on DL (0.5571 vs. 0.5760) and BEIR (0.5419 vs. 0.5513). **Fourth**, selection is critical once a pool of documents is available. On BEIR, agreement- and rank-based selection (0.5330, 0.5328) sit only marginally above Cold, whereas relevance-based selection reaches 0.5513; the largest gains come from identifying useful documents within the pool, not pooling alone. **Fifth**, effectiveness depends on both the selector and the candidate pool. Table 3 and Figure 1 show the same relevance-based selector applied to the original query alone (BM25@100) reaches only 0.540/0.535 on DL/BEIR, versus 0.576/0.551 when drawing from the reformulator portfolio. Much of the gain is recovered with small portfolios, while larger ones add robustness; a gap to the oracle (using qrels as documents) remains across all sizes, suggesting future gains lie in better selection rather than larger pools. **Finally**, the pattern is robust across generation backbones. Table 4 shows relevance-based selection achieves the strongest average under both `DeepSeek-V3` and `Llama-3.3-70B`; though the ordering

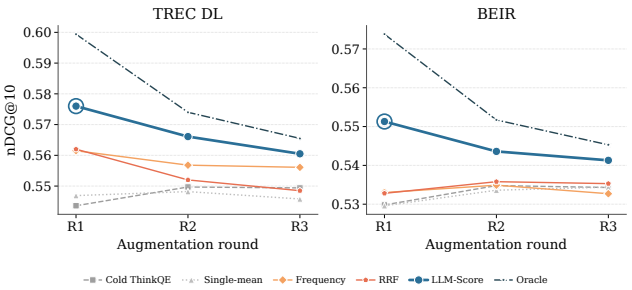

Figure 2: Iteration dynamics across 3 rounds, averaged nDCG@10 over TREC DL (left) and BEIR (right).

of agreement- and rank-based selectors varies slightly, the overall trend holds, indicating the effect is not tied to a particular LLM.

RQ3: Do additional retrieval-generation rounds provide further benefit? We examine whether iterative retrieval-generation pipelines still help once high-quality conditioning documents have been identified. This is important because many recent expansion methods derive effectiveness from multiple rounds, yet it is unclear whether these gains come from repeated reformulation or simply from improved access to useful documents.

We observe that **first**, strong document selection consistently achieves its best performance in the first round. Relevance-based selection reaches its highest BEIR-averaged nDCG@10 after a single expansion round (0.5513), before declining to 0.5436 and 0.5413 in the second and third rounds, respectively. The same pattern is observed on the DL average and across individual datasets. Notably, a single expansion round conditioned on carefully selected documents outperforms multiple rounds of iterative expansion operating over weaker document sets. The decline from the first to the third round on BEIR is approximately 0.010 nDCG@10, which is comparable in magnitude to the differences separating several of the strongest strategies evaluated in RQ2. The oracle pool follows a similar but steeper trajectory, suggesting iterative retrieval mainly recovers documents strong selection has already identified rather than uncovering new information. **Second**, iterative expansion helps little when the initial document set is weak. On BEIR, Cold ThinkQE improves only marginally across rounds (0.5298, 0.5348, 0.5343), and Single-mean is similarly flat, while agreement- and rank-based selection stay flat or decline. These results indicate the primary benefit of iteration is acquiring better conditioning documents; once high-quality documents are supplied, further cycles add little and may introduce noise that degrades effectiveness.

5 Concluding Remarks

This paper examined the role of conditioning documents in LLM-based query expansion. We showed that different reformulation methods retrieve distinct relevant documents, and proposed EviQE, which separates document selection from generation through portfolio-based pooling and selection. Across TREC DL and BEIR, selecting high-quality conditioning documents consistently improves effectiveness over direct reformulation, cold-start expansion, and single-source seeding, while reducing the benefit of additional iterations. These findings indicate that document selection is a critical yet underexplored component of query expansion, and that much of the benefit attributed to iterative pipelines may stem from better conditioning documents rather than repeated reformulation.

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] 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

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

[3] 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

[4] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 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* (Boston, MA, USA) *(SIGIR '09)*. Association for Computing Machinery, New York, NY, USA, 758–759. doi:10.1145/1571941.1572114

[5] 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

[6] 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).

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

[8] 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.

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

[10] Guglielmo Faggioli et al. 2023. Perspectives on Large Language Models for Relevance Judgment. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR)*. ACM.

[11] 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

[12] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise Zero-Shot Dense Retrieval without Relevance Labels. arXiv:2212.10496 [cs.IR] https://arxiv.org/abs/2212.10496

[13] Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. *arXiv preprint arXiv:2305.03653* (2023).

[14] 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

[15] 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

[16] 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

[17] Yibin Lei, Tao Shen, and Andrew Yates. 2025. ThinkQE: Query Expansion via an Evolving Thinking Process. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (Eds.). Association for Computational Linguistics, Suzhou, China, 17772–17781. doi:10.18653/v1/2025.findings-emnlp.965

[18] Sean MacAvaney and Arman Cohan. 2023. One-Shot Labeling for Automatic Relevance Estimation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM.

[19] 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.

[20] Zhen Qin et al. 2024. Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*. Association for Computational Linguistics.

[21] 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.

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

[23] 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).

[24] 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

[25] Weiwei Sun et al. 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)*. Association for Computational Linguistics.

[26] Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2024. LLM Judges for Relevance Grading in Information Retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM.

[27] 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).

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

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

[30] 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