QA-Expand: Multi-Question Answer Generation for Enhanced Query Expansion in Information Retrieval

Anonymous ACL submission

Abstract

Query expansion is widely used in Information Retrieval (IR) to improve search outcomes by enriching queries with additional contextual information. Although recent Large Language Model (LLM) based methods generate pseudo-relevant content and expanded terms via multiple prompts, they often yield repetitive, narrow expansions that lack the diverse context needed to retrieve all relevant information. In this paper, we introduce ***QA-Expand, a novel and effective framework for query expansion.*** It first generates multiple relevant questions from the initial query and subsequently produces corresponding pseudo-answers as surrogate documents. A feedback model further filters and rewrites these answers to ensure only the most informative augmentations are incorporated. Extensive experiments on benchmarks such as *BEIR* and *TREC* demonstrate that *QA-Expand* enhances retrieval performance by up to 13% over state-of-the-art methods, offering a robust solution for modern retrieval challenges.

1 Introduction

Query expansion is widely used in Information Retrieval (IR) for effectively improving search outcomes by enriching the initial query with additional contextual information (Carpineto and Romano, 2012; Azad and Deepak, 2019a; Jagerman et al., 2023). Traditional methods as Pseudo-Relevance Feedback (PRF) expand queries by selecting terms from top-ranked documents (Robertson, 1990; Jones et al., 2006; Lavrenko and Croft, 2017). While these conventional approaches have been successful to some extent, their reliance on static term selection limits the scope of expansion (Roy et al., 2016; Imani et al., 2019).

In recent years, Large Language Models (LLMs) have enabled dynamic query rewriting techniques that overcome traditional limitations by harnessing their generative ability (Zhao et al., 2023; Ye et al., 2023; Liu and Mozafari, 2024; Lei et al., 2024; Seo et al., 2024; Chen et al., 2024). For instance, *Q2D* (Wang et al., 2023) expands queries with pseudo-documents generated via few-shot prompting, while *Q2C* (Jagerman et al., 2023) uses Chain-of-Thought (CoT) prompting (Wei et al., 2022) for reformulation. Moreover, *GenQREnsemble* (Dhole and Agichtein, 2024) concatenates multiple keyword sets produced through zero-shot paraphrasing with the original query, and *GenQRFusion* (Dhole et al., 2024) retrieves documents for each keyword set and fuses the rankings.

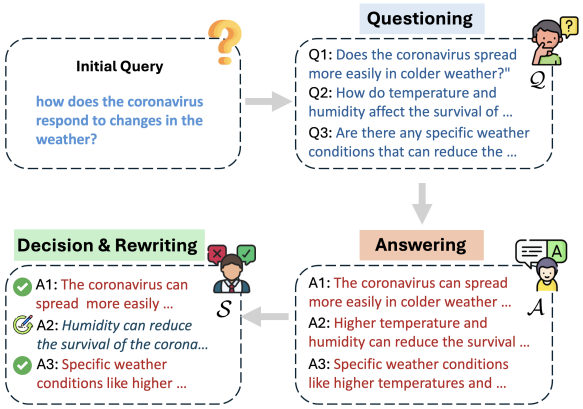

Figure 1: **The overview of the novel *QA-Expand* framework.** Given an initial query, the framework generates diverse relevant questions, produces corresponding pseudo-answers, and selectively rewrites and filters relevant answers to enhance query expansion.

Despite these advances, several significant challenges remain: ① simplistic prompt variations yield repetitive, narrowly focused expansions that miss the full range of contextual nuances; ② many approaches lack a dynamic evaluation mechanism, leading to redundant or suboptimal term inclusion; and ③ these methods do not reformulate the query into distinct questions with corresponding answers, limiting their ability to capture diverse, insightful facets of the underlying information need.

To overcome these limitations, we propose *QA-Expand*, a novel framework that leverages Large

Language Models (LLMs) to generate diverse question-answer pairs from an initial query. Specifically, *QA-Expand* first generates multiple relevant questions derived from the initial query and subsequently produces corresponding pseudo-answers that serve as surrogate documents to enrich the query representation. A feedback model is furthermore integrated to selectively rewrite and filter these generated answers, ensuring that the final query augmentation robustly captures a multi-faceted view of the underlying information need.

Extensive experiments on four datasets from *BEIR Benchmark* (Thakur et al., 2021) and two datasets from the *TREC Deep Learning Passage* 2019 and 2020 (Craswell et al., 2020) demonstrate that *QA-Expand* significantly outperforms existing query expansion techniques. Our contributions include: (1) *a novel paradigm* that reformulates the query into multiple targeted questions and generates corresponding pseudo-answers to capture diverse aspects of the information need; (2) *a dynamic feedback model* that selectively rewrites and filters only the most informative pseudo-answers for effective query augmentation; and (3) *comprehensive empirical validation* confirming the robustness and superiority of our approach.[1]

2 Methodology

In this section, we detail our proposed *QA-Expand* framework. An overview of the *QA-Expand* framework is provided in Figure 1.

2.1 Multiple Question Generation

Given an initial query Q, a single inference call is made to an LLM using a fixed prompt P to generate a set of diverse questions relevant to initial query. This process is formalized as:

$$\mathcal{Q} = \{q_1, q_2, \ldots, q_N\} = G_{\mathcal{Q}}(Q, P), \quad (1)$$

where $G_{\mathcal{Q}}$ denotes the question generation module and N is the number of generated questions. Each q_i is designed to capture a distinct aspect of the information need expressed in Q.

2.2 Pseudo-Answer Generation

For each generated question $q_i \in \mathcal{Q}$, the answer generation module subsequently produces a corresponding pseudo-answer. This module generates an answer for each question in $\mathcal{Q}$. This results in a complete set of pseudo-answers:

$$\mathcal{A} = \{a_1, a_2, \ldots, a_N\} = G_{\mathcal{A}}(\mathcal{Q}), \quad (2)$$

where $G_{\mathcal{A}}$ denotes the answer generation process implemented via an LLM. This design ensures that all generated questions are paired with an answer, providing a comprehensive candidate set for subsequent evaluation.

2.3 Feedback-driven Rewriting and Selection

After generating the pseudo-answers, the feedback module $G_{\mathcal{S}}$ processes the complete set of question-answer pairs $\{(q_i, a_i)\}_{i=1}^{N}$ in the context of the initial query Q and directly produces the refined pseudo-answer set:

$$\mathcal{S} = G_{\mathcal{S}}(\{(q_i, a_i)\}_{i=1}^{N}, Q). \quad (3)$$

Here, $G_{\mathcal{S}}$ denotes the selective rewriting and filtering operation implemented via an LLM. In this process, any refined pseudo-answer deemed irrelevant or too vague is omitted from $\mathcal{S}$. Thus, the final set of refined pseudo-answers can be represented as:

$$\mathcal{S} = \{a'_1, a'_2, \ldots, a'_j\}, \quad \text{with } 0 \leq j \leq N. \quad (4)$$

Finally, the refined pseudo-answers $\mathcal{S}$ are integrated with the initial query Q using various aggregation strategies (e.g., sparse concatenation, dense weighted fusion, and Reciprocal Rank Fusion), as detailed in Section 3.2.[2]

3 Experiments

3.1 Setup

Our experimental setup includes a description of the datasets, model specifications, and the baseline methods used for comparison in our framework.

Datasets. We evaluate QA-Expand on two benchmark collections: (1) *BEIR Benchmark* (Thakur et al., 2021) and (2) *TREC Passage Datasets* (Craswell et al., 2020). Specifically, for *BEIR Benchmark*, we select four frequently used datasets: *webis-touche2020*, *scifact*, *trec-covid-beir*, and *dbpedia-entity*. For *TREC Datasets*, we employ the *Deep Learning Passage Tracks* from 2019 and 2020, which consist of large-scale passage collections to ensure that our approach performs well in challenging retrieval scenarios. [3]

[1] Background is detailed in Appendix A.

[2] Prompts are detailed in Appendix B.

[3] Statistical description of the datasets is detailed in Appendix C.

LLM and Retrieval Models. For generating the question-answer pairs in *QA-Expand*, we utilize *Qwen2.5-7B-Instruct Model*[4] (Team, 2024), which is a high-performance language model and, for the retrieval task, we employ *multilingual-e5-base*[5] (Wang et al., 2024) to encode both queries and documents into dense representations. Additionally, we incorporate *BM25* (Robertson et al., 2009) as a sparse retrieval baseline, specifically using BM25s[6] (Lù, 2024), a pure-Python implementation that leverages Scipy (Virtanen et al., 2020) sparse matrices for fast, efficient scoring.

Baselines and Our Approach. We compare *QA-Expand* with standard retrieval baselines and query expansion methods. Retrieval baselines include *BM25* for sparse retrieval and *multilingual-e5-base* for dense retrieval using cosine similarity. We also evaluate query expansion methods such as *Q2D* (Wang et al., 2023), which generates pseudo-documents via few-shot prompting, and *Q2C* (Jagerman et al., 2023), which uses chain-of-thought guided reformulation. In addition, we compare with *GenQR-based* methods (Dhole and Agichtein, 2024; Dhole et al., 2024) that generate 10 prompt-based keyword sets, with one variant concatenating these keywords with the original query and the other retrieving documents for each set and fusing the rankings. All baselines use their original settings. In contrast, our *QA-Expand* enriches each query by generating 3 distinct while diverse questions and corresponding refined pseudo-answers—a configuration chosen to balance diversity and relevance by capturing multiple facets of the query without excessive redundancy.

3.2 Implementation Details

Sparse Query Aggregation. In the sparse retrieval setting, following previous work (Wang et al., 2023; Jagerman et al., 2023), we replicate the initial query Q three times and append all refined pseudo-answers a'_i. Specifically, let $Q_i = Q$ for $i = 1, 2, 3$. The expanded query is formulated as:

$$Q^*_{\text{sparse}} = \sum_{i=1}^{3} Q_i + \sum_{j=1}^{|\mathcal{S}|} a'_j, \tag{5}$$

where the "+" operator denotes the concatenation of query terms (with [SEP] tokens as separators).

Dense Query Aggregation. For dense retrieval, let $\text{emb}(Q)$ be the embedding of the initial query and $\text{emb}(a'_i)$ the embedding of each refined pseudo-answer. Following previous work in weighted query aggregations (Seo et al., 2024), which employed a weight of 0.7 for the initial query embedding, we adopted the same weighting scheme and compute the final query embedding Q^*_{dense}:

$$Q^*_{\text{dense}} = 0.7 \cdot \text{emb}(Q) + 0.3 \cdot \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \text{emb}(a'_i). \tag{6}$$

Reciprocal Rank Fusion (RRF). In the RRF setting (Cormack et al., 2009), each refined pseudo-answer a'_i is used to form an individual expanded query Q^*_i. For each document d, let $r_{i,d}$ denote its rank when retrieved with Q^*_i. The final score for d is computed as:

$$\text{score}(d) = \sum_{i=1}^{|\mathcal{S}|} \frac{1}{k + r_{i,d}}, \tag{7}$$

where k is a constant (e.g., $k = 60$) to dampen the influence of lower-ranked documents. Documents are then re-ranked based on their aggregated scores.[7]

3.3 Main Results

In our experiments on both sparse and dense retrieval settings (see Table 1), we found that while methods such as *GenQREnsemble*, *Q2C*, and *Q2D* yield incremental improvements through query reformulations, each exhibits notable shortcomings. *GenQREnsemble* uses multiple prompt configurations to produce pseudo-relevant term expansions, yet its repeated and narrowly focused outputs often miss the full spectrum of user intent. Similarly, *Q2C* leverages chain-of-thought (CoT) reasoning but tends to generate repetitive expansions with limited contextual diversity, and although *Q2D* produces pseudo-documents that better capture the underlying information need, it falls short in filtering out less informative content. In contrast, our *QA-Expand* framework reformulates the query into diverse targeted questions and generates corresponding pseudo-answers that are dynamically evaluated, resulting in a 13% improvement in average retrieval performance.

[4] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

[5] https://huggingface.co/intfloat/multilingual-e5-base

[6] https://github.com/xhluca/bm25s

[7] Algorithm of *QA-Expand* is detailed in Appendix D.

Methods	BEIR Benchmark (nDCG@10)					TREC DL'19			TREC DL'20		
	Webis	SciFact	TREC-COVID	DBpedia	Avg. Score	nDCG@10	R@1000	Avg. Score	nDCG@10	R@1000	Avg. Score
Sparse Results											
BM25	0.2719	0.6694	0.5868	0.2831	0.4528	0.4239	0.3993	0.4116	0.4758	0.4240	0.4500
Q2C (2023)	0.3546	0.6876	0.6954	0.3252	0.5157	0.5439	0.4814	0.5127	0.5357	0.4941	0.5149
Q2D (2023)	0.3679	0.6794	0.6957	**0.3378**	0.5202	0.5732	0.4890	0.5311	0.5486	0.4958	0.5222
GenQREnsemble (2024)	0.2887	0.5560	0.5104	0.2302	0.3963	0.4109	0.4110	0.4110	0.4261	0.4163	0.4207
QA-Expand* (Sparse, Ours)	**0.3919***	**0.6965***	**0.7050***	0.3273*	**0.5302***	**0.5811***	**0.4932***	**0.5372***	**0.5803***	**0.5000***	**0.5402***
Dense Results											
E5-Base	0.1786	0.6924	0.7098	0.4002	0.4953	0.7020	0.5185	0.6103	0.7029	0.5648	0.6339
Q2C (2023)	0.1841	0.7028	0.7238	0.4250	0.5112	0.5517	0.4891	0.5204	0.7084	0.5715	0.6400
Q2D (2023)	**0.1931**	0.7108	0.7284	0.4229	0.5133	0.7472	**0.5565**	**0.6519**	0.6971	0.5799	0.6385
QA-Expand* (Dense, Ours)	0.1911*	**0.7147***	**0.7342***	**0.4278***	**0.5387***	**0.7476***	0.5527*	0.6502*	**0.7184***	**0.5831***	**0.6508***
RRF Fusion (BM25) Results											
GenQRFusion (2024)	**0.3815**	0.6518	0.6594	0.2726	0.4913	0.4418	0.4205	0.4312	0.4375	0.4654	0.4515
QA-Expand* (RRF, Ours)	0.3533	**0.6777***	**0.6698***	**0.3009***	**0.5004***	**0.5048***	**0.4734***	**0.4891***	**0.5211***	**0.4795***	**0.5003***

Table 1: *Combined retrieval performance on BEIR Benchmark (nDCG@10) and TREC DL'19/TREC DL'20 (nDCG@10 / R@1000). For BEIR, the Avg. column is the average across Webis, SciFact, TREC-COVID, and DBpedia*. For TREC DL, the Avg. Score is computed as the average of nDCG@10 and R@1000. Bold indicates the best score and underline indicates the second-best score. * denotes significant improvements (paired t-test with Holm-Bonferroni correction, $p < 0.05$) over the average baseline value for the metric.

In the fusion-based retrieval scenario, conventional methods such as *GenQRFusion* typically generate candidates through up to ten separate prompts and then fuse the resulting rankings using Reciprocal Rank Fusion (RRF). Although this approach is intended to capture a wide range of query facets, it often aggregates redundant or low-quality candidates, resulting in an overall ineffective expansion. Our *QA-Expand* framework, on the other hand, employs a more discerning selection process prior to fusion. Our method integrates only those expansions that robustly encapsulate the multifaceted nature of the initial query by leveraging a dedicated evaluation module to filter out inferior pseudo-answers. This targeted fusion strategy minimizes computational overhead while delivering significantly improved retrieval performance, as evidenced by our experimental results.

3.4 Ablation Study and Analysis

To evaluate the effectiveness of the evaluation module, we conducted an ablation study on two datasets. Table 2 compares the full *QA-Expand* framework with a variant that omits the evaluation module. The results show that including the evaluation module improves the average score by effectively filtering out redundant and less informative pseudo-answers, ensuring that only high-quality expansions contribute to query augmentation.

Furthermore, the evaluation module not only boosts overall performance but also enhances robustness. Without it, performance variability increases and more noise from less relevant pseudo-answers is observed, whereas the refined feedback mechanism maintains stable and superior retrieval effectiveness across diverse datasets. These findings highlight the importance of dynamically selecting high-quality expansions to capture the multifaceted nature of user intent. Notably, even the variant without the evaluation module outperforms other baselines, as shown in Table 1.

Methods	Feedback	BEIR	TREC DL'19	TREC DL'20
BM25	w/o feedback	0.5266	0.5342	0.5373
	w feedback	**0.5302**	**0.5372**	**0.5402**
Dense	w/o feedback	0.5115	0.6404	0.6474
	w feedback	**0.5387**	**0.6502**	**0.6508**
BM25/RRF	w/o feedback	**0.5099**	0.4766	0.5001
	w feedback	0.5004	**0.4891**	**0.5003**

Table 2: *Combined average retrieval performance on BEIR Benchmark and TREC DL datasets, with and without feedback*. Scores are averaged over four BEIR datasets and computed separately for TREC DL'19 and DL'20. Bold values denote the best performance.

4 Conclusion

In this paper, we present our novel framework *QA-Expand* which addresses query expansion by generating diverse question-answer pairs and employing a feedback model for selective rewriting and filtering. Our approach yields significant performance gains and better captures the multifaceted nature of user intent. Experimental results on BEIR and TREC benchmarks demonstrate the effectiveness and robustness of *QA-Expand*.

5 Limitations

One limitation is the persistence of residual noise and redundancy in the expanded queries. Although our feedback module is designed to filter out irrelevant or repetitive pseudo-answers, some less informative content may still be included, particularly for queries with ambiguous or complex information needs. Such residual noise can degrade retrieval precision by diluting the core intent of the initial query. Further research is needed to develop more robust filtering methods that can better discern and eliminate spurious information. Addressing this issue is an important direction for future work, as it could significantly improve the effectiveness of the query expansion process.

References

Hiteshwar Kumar Azad and Akshay Deepak. 2019a. A novel model for query expansion using pseudo-relevant web knowledge. *arXiv preprint arXiv:1908.10193*.

Hiteshwar Kumar Azad and Akshay Deepak. 2019b. Query expansion techniques for information retrieval: a survey. *Information Processing & Management*, 56(5):1698–1735.

Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. *Acm Computing Surveys (CSUR)*, 44(1):1–50.

Xinran Chen, Xuanang Chen, Ben He, Tengfei Wen, and Le Sun. 2024. Analyze, generate and refine: Query expansion with llms for zero-shot open-domain qa. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11908–11922.

Vincent Claveau. 2020. Query expansion with artificially generated texts. *arXiv preprint arXiv:2012.08787*.

Gordon V Cormack, Charles LA 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*, pages 758–759.

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*.

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

Kaustubh D Dhole, Ramraj Chandradevan, and Eugene Agichtein. 2024. Generative query reformulation using ensemble prompting, document fusion, and relevance feedback. *arXiv preprint arXiv:2405.17658*.

Ayyoob Imani, Amir Vakili, Ali Montazer, and Azadeh Shakery. 2019. Deep neural networks for query expansion using word embeddings. In *Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part II 41*, pages 203–210. Springer.

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

Pengyue Jia, Yiding Liu, Xiangyu Zhao, Xiaopeng Li, Changying Hao, Shuaiqiang Wang, and Dawei Yin. 2023. Mill: Mutual verification with large language models for zero-shot query expansion. *arXiv preprint arXiv:2310.19056*.

Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. 2006. Generating query substitutions. In *Proceedings of the 15th international conference on World Wide Web*, pages 387–396.

Ivica Kostric and Krisztian Balog. 2024. A surprisingly simple yet effective multi-query rewriting method for conversational passage retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2271–2275.

Victor Lavrenko and W Bruce Croft. 2017. Relevance-based language models. In *ACM SIGIR Forum*, volume 51, pages 260–267. ACM New York, NY, USA.

Yibin Lei, Yu Cao, Tianyi Zhou, Tao Shen, and Andrew Yates. 2024. Corpus-steered query expansion with large language models. *arXiv preprint arXiv:2402.18031*.

Hang Li, Ahmed Mourad, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. 2023. Pseudo relevance feedback with deep language models and dense retrievers: Successes and pitfalls. *ACM Transactions on Information Systems*, 41(3):1–40.

Minghan Li, Honglei Zhuang, Kai Hui, Zhen Qin, Jimmy Lin, Rolf Jagerman, Xuanhui Wang, and Michael Bendersky. 2024. Can query expansion improve generalization of strong cross-encoder rankers? In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2321–2326.

Jie Liu and Barzan Mozafari. 2024. Query rewriting via large language models. *arXiv preprint arXiv:2403.09060*.

Xing Han Lù. 2024. Bm25s: Orders of magnitude faster lexical search via eager sparse scoring. *arXiv preprint arXiv:2407.03618*.

Iain Mackie, Shubham Chatterjee, and Jeffrey Dalton. 2023. Generative and pseudo-relevant feedback for sparse, dense and learned sparse retrieval. *arXiv preprint arXiv:2305.07477*.

Shahrzad Naseri, Jeffrey Dalton, Andrew Yates, and James Allan. 2021. Ceqe: Contextualized embeddings for query expansion. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part I 43*, pages 467–482. Springer.

Hai-Long Nguyen, Tan-Minh Nguyen, Duc-Minh Nguyen, Thi-Hai-Yen Vuong, Ha-Thanh Nguyen, and Xuan-Hieu Phan. 2024. Exploiting llms' reasoning capability to infer implicit concepts in legal information retrieval. *arXiv preprint arXiv:2410.12154*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Stephen E Robertson. 1990. On term selection for query expansion. *Journal of documentation*, 46(4):359–364.

Dwaipayan Roy, Debjyoti Paul, Mandar Mitra, and Utpal Garain. 2016. Using word embeddings for automatic query expansion. *arXiv preprint arXiv:1606.07608*.

Wonduk Seo, Haojie Zhang, Yueyang Zhang, Changhao Zhang, Songyao Duan, Lixin Su, Daiting Shi, Jiashu Zhao, and Dawei Yin. 2024. Gencrf: Generative clustering and reformulation framework for enhanced intent-driven information retrieval. *arXiv preprint arXiv:2409.10909*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. 2020. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Fanghua Ye, Meng Fang, Shenghui Li, and Emine Yilmaz. 2023. Enhancing conversational search: Large language model-aided informative query rewriting. *arXiv preprint arXiv:2310.09716*.

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*, pages 1872–1883.

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*.

A Appendix A. Background

Query Expansion with LLM. Let Q denote the initial query and G be a Large Language Model (LLM) used for generation. Query expansion enhances retrieval by enriching Q with additional context (Azad and Deepak, 2019b; Claveau, 2020; Naseri et al., 2021; Jia et al., 2023). Two predominant LLM-based strategies have emerged: (1) *Pseudo-Document Generation*, where G produces a surrogate document D or an expanded query Q^* using a prompt P to capture latent information (Wang et al., 2023; Jagerman et al., 2023; Zhang et al., 2024), and (2) *Term-Level Expansion*, where G generates a set of terms $T = \{t_1, t_2, \ldots, t_M\}$ that reflect diverse aspects of Q (Dhole and Agichtein, 2024; Dhole et al., 2024; Li et al., 2024; Nguyen et al., 2024).

Retrieval with Expanded Queries. In query expansion, Information Retrieval (IR) integrates the initial query with generated augmentations using various strategies. For *sparse retrieval*, the common method concatenates multiple copies of the initial query with generated terms to reinforce core signals (Wang et al., 2023; Zhang et al., 2024). In *dense retrieval*, one strategy directly combines the query and its expansions into a unified embedding (Li et al., 2023; Wang et al., 2023), while another fuses separate embeddings from each component (Seo et al., 2024; Kostric and Balog, 2024). Additionally, *Reciprocal Rank Fusion (RRF)* aggregates rankings from individual expanded queries by inversely weighting document ranks (Mackie et al., 2023).

B Appendix B. Prompts

Prompt for Multiple Question Generation

You are a helpful assistant. Based on the following query, generate 3 possible related questions that someone might ask.
Format the response as a JSON object with the following structure:

```
{"question1":"First question ..."
"question2":"Second question ..."
"question3":"Third question ..."}
```

Only include questions that are meaningful and logically related to the query. Here is the query: {}

Prompt for Pseudo-Answer Generation

You are a knowledgeable assistant. The user provides 3 questions in JSON format. For each question, produce a document style answer. Each answer must: Be informative regarding the question. Return all answers in JSON format with the keys `answer1`, `answer2`, and `answer3`. For example:

```
{"answer1": "...",
"answer2": "...",
"answer3": "..."}
```

Text to answer: {}

Prompt for Feedback-driven Rewriting and Selection

You are an evaluation assistant. You have an initial query and answers provided in JSON format. Your role is to check how relevant and correct each answer is. Return only those answers that are relevant and correct to the initial query. Omit or leave blank any that are incorrect, irrelevant, or too vague. If needed, please rewrite the answer in a better way.
Return your result in JSON with the same structure:

```
{"answer1": "Relevant/correct...",
"answer2": "Relevant/correct...",
"answer3": "Relevant/correct..."}
```

If an answer is irrelevant, do not include it at all or leave it empty. Focus on ensuring the final JSON only contains the best content for retrieval. Here is the combined input (initial query and answers): {}

C Appendix C. Dataset Details

Dataset	Test Queries	Corpus
Webis	49	382,545
SciFact	300	5,183
TREC-COVID	50	171,332
DBpedia-Entity	400	4,635,922
Trec DL'19 Passage	43	8,841,823
Trec DL'20 Passage	54	8,841,823

Table 3: Test Queries and Corpus Sizes for the Different Datasets from *BEIR Benchmark* and *TREC Track*.

D Appendix D. Algorithm

Algorithm 1 *QA-Expand*: Query Expansion via Question-Answer Generation

Require: Initial query Q, LLM models: Question Generator $G_{\mathcal{Q}}$, Answer Generator $G_{\mathcal{A}}$, Feedback Filter $G_{\mathcal{S}}$, Aggregation Strategy Agg

Ensure: Expanded query Q^*

// Step 1: Multiple Question Generation

$\mathcal{Q} \leftarrow G_{\mathcal{Q}}(Q, P)$ ▷ Generate a set of diverse questions $\{q_1, q_2, \ldots, q_N\}$ from Q

// Step 2: Pseudo-Answer Generation

$\mathcal{A} \leftarrow G_{\mathcal{A}}(\mathcal{Q})$ ▷ Generate pseudo-answers concurrently for all questions, yielding $\mathcal{A} = \{a_1, a_2, \ldots, a_N\}$

// Step 3: Feedback-driven Rewriting and Selection

$\mathcal{S} \leftarrow G_{\mathcal{S}}(\{(q_i, a_i)\}_{i=1}^{N}, Q)$ ▷ Refine and filter to obtain $\mathcal{S} = \{a'_1, a'_2, \ldots, a'_j\}$ with $0 \leq j \leq N$

// Retrieval via Diverse Aggregation Methods

if Method = Sparse **then**

  Compute: $Q^*_{\text{sparse}} = \sum_{i=1}^{3} Q_i + \sum_{j=1}^{|\mathcal{S}|} a'_j$

else if Method = Dense **then**

  Compute: $Q^*_{\text{dense}} = 0.7 \cdot \text{emb}(Q) + 0.3 \cdot \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \text{emb}(a'_i)$

else if Method = RRF **then**

  for each document d **do**

    Compute: $\text{score}(d) = \sum_{i=1}^{|\mathcal{S}|} \frac{1}{k + r_{i,d}}$

  end for

end if

return Ranked documents