Augmenting LLM-based Zero-Shot Retrieval with Listwise Ranking

Anonymous ACL submission

Abstract

Modern IR systems rely on retrieval-reranking pipelines that combine efficient first-stage retrieval with accurate yet expensive second-stage ranking. While distilling ranking knowledge into retrievers makes retrieval effective, existing approaches require supervision and are unsuitable for zero-shot settings. We propose LRaR, a zero-shot retrieval framework that integrates discriminative listwise ranking signals into corpus-wide retrieval using a single LLM call. LRaR jointly performs query augmentation and extracts ranking preferences from a small set of initial candidates, mapping them into a retrieval-compatible representation without training or relevance annotations. Experiments on DL19, DL20, and BEIR show that LRaR consistently improves LLM-based retrieval across diverse LLMs, matching or outperforming reranking over 100 documents using signals from only the top 20, while being substantially more efficient.

1 Introduction

To balance efficiency and effectiveness, modern information retrieval (IR) systems commonly adopt a "retrieval-reranking pipeline" (Nogueira et al., 2019; Pradeep et al., 2021). A first-stage retriever, usually achieved by a lightweight bi-encoder, is used to efficiently recall a small set of candidate documents, which are then refined by a more expressive second-stage ranker, by a cross-encoder, that focuses on precise relevance estimation. Because second-stage rankers operate on a limited set of candidates, they can model richer query–document interactions and capture fine-grained relevance distinctions, typically achieving better accuracy (Nogueira et al., 2020; Ma et al., 2024). As a result, if the knowledge encoded in powerful rankers can be incorporated into the construction of retrievers, the effectiveness of retrieval can be significantly improved while preserving efficiency, which is usually achieved by distilling listwise relevance signals from strong rerankers into the retriever during training (Ren et al., 2021; Shen et al., 2023).

Despite its effectiveness in supervised settings, such distillation-based approaches crucially rely on large amounts of relevance annotations or logged interaction data, and therefore cannot be directly applied to zero-shot retrieval, a setting that is critical for scientific search, domain adaptation, and low-resource or rapidly evolving information environments. In these settings, retrievers must generalize to new domains and tasks without any task-specific training signals, making supervised ranker-to-retriever distillation unsuitable.

This limitation has become especially salient in the era of large language models (LLMs). Recent advances in LLMs have enabled a new paradigm of LLM-based retrieval (Mackie et al., 2023; Gao et al., 2022; Wang et al., 2023b; Zhang et al., 2024), where a model's internal knowledge, reasoning ability, and generative capacity are leveraged to augment or reformulate user queries, which are then matched against the corpus using a general-purpose retrieval model such as BM25 (Robertson et al., 1995) or dense embedding similarity (Izacard et al., 2022; Karpukhin et al., 2020). While this paradigm offers strong zero-shot performance and remarkable flexibility, it remains largely generative in nature and lacks explicit mechanisms to incorporate discriminative relevance signals comparable to those provided by listwise ranking models.

Empowered by LLMs' multi-task capabilities and motivated by the central role of listwise ranking knowledge in defining retrieval-oriented relevance, we propose LRaR (LLM-based Ranking-augmented Retrieval), a zero-shot retrieval framework that bridges the long-standing gap between discriminative ranking signals and efficient corpus-wide retrieval. Built upon query augmentation, LRaR uses the single LLM call to also extract listwise relevance preferences from initial candidates,

which are mapped into a retrieval-compatible query representation, enabling ranking-aware, corpus-wide retrieval without supervision. This induces a dual-perspective retrieval process that jointly captures generative query intent and discriminative relevance ordering.

Extensive experiments show that LRaR yields consistent gains on DL19, DL20, and BEIR across diverse LLM backbones. Notably, although LRaR leverages ranking signals extracted from only the top-20 candidates, it can match or surpass conventional reranking over the top-100 documents, while being significantly more efficient. Additional analyses indicate that LRaR provides information complementary to generative query augmentation.

2 Related Work

LLM-based Zero-Shot Retrieval. Recent work has explored leveraging LLMs to enable zero-shot retrieval without task-specific supervision, motivated by their strong generalization and reasoning abilities (Mackie et al., 2023; Lei et al., 2024; Zhang et al., 2024). Most approaches use LLMs to transform or augment the original query, often by generating hypothetical documents or answer-like passages (Gao et al., 2022; Wang et al., 2023b; Shen et al., 2024), explanations (Jagerman et al., 2023), or expanded term sets (Mackie et al., 2023; Jagerman et al., 2023), which are then matched against the corpus using standard retrievers. In contrast, our work goes beyond purely generative augmentation by extracting discriminative listwise ranking signals from LLMs and explicitly integrating them into corpus-wide retrieval.

Building Retriever with Reranker. A line of work has shown that retrievers can be substantially improved by distilling listwise relevance signals produced by strong rerankers over a shared set of candidate documents, as such signals capture fine-grained, comparative notions of relevance that go beyond pointwise supervision. Representative approaches include RocketQA-V2 (Ren et al., 2021), AR2 (Zhang et al., 2022), and SimLM (Wang et al., 2023a), which leverage cross-encoder or reranker outputs to supervise retriever training and align retrieval representations with ranking-oriented objectives. While highly effective, these methods fundamentally rely on supervised training and annotated or logged data. In contrast, our work transfers listwise ranking knowledge to retrieval in a fully zero-shot manner, using LLMs to induce ranking-aware retrieval representations during inference without any training or relevance annotations.

3 Method

LRaR is a zero-shot retrieval framework that augments LLM-based retrieval with discriminative listwise ranking signals, enabling ranking-aware corpus-wide retrieval using a single LLM call.

Retrieval via Query Augmentation. We start from the standard paradigm of LLM-based retrieval, where an LLM is used to augment or reformulate the input query prior to retrieval. Given a query q and a document corpus $\mathcal{C}$, an LLM-based query augmentation model $\mathcal{L}$ produces an augmented query $\tilde{q}^{\text{gen}} = \mathcal{L}(q)$, which is then used by a first-stage retriever R (e.g., BM25 or dense retrieval) to obtain an initial candidate set: $\mathcal{D}^{(0)} = R(\tilde{q}^{\text{gen}}, \mathcal{C})$. This paradigm leverages the generative and reasoning capabilities of LLMs, but remains purely generative and does not explicitly incorporate discriminative relevance signals.

Listwise Ranking. To inject discriminative relevance information, LRaR applies an LLM-based listwise ranker over a small candidate set. Given the original query q and the initial candidates $\mathcal{D}^{(0)}$, the LLM jointly produces: $\pi, \mathbf{F} = \mathcal{L}(q, \mathcal{D}^{(0)})$, where π is a ranked permutation of documents and $\mathbf{F}$ denotes listwise ranking feedback derived from comparative reasoning over the candidate set. In conventional pipelines, π would be used solely for reranking $\mathcal{D}^{(0)}$; in contrast, LRaR transforms this listwise ranking signal into a retrieval-compatible representation. However, directly fusing the reranking order π with retrieval scores is limited: π is defined only over the truncated set $\mathcal{D}^{(0)}$ and cannot assign corpus-comparable scores to unseen documents, leading to a mismatch. LRaR instead converts listwise preferences into a retrieval-compatible query representation, allowing discriminative ranking signals to guide corpus-wide search.

Ranking-Augmented Retrieval. The core idea of LRaR is to map listwise ranking preferences into an augmented query that can be used for corpus-wide retrieval. Based on the ranking output, we identify a subset of documents deemed highly relevant according to $\mathbf{F}$. We then construct a ranking-augmented query: $\tilde{q}^{\text{rank}} = q \oplus \bigoplus_{i=1}^{k} \underbrace{d_i \oplus \cdots \oplus d_i}_{w_i \text{ times}}$, where $\{d_i\}_{i=1}^{k}$ are documents marked as relevant by $\mathbf{F}$, r_i is the rank of d_i in π,

and $w_i = \left\lceil \frac{100}{r_i} \right\rceil$ controls the contribution of each document based on its relative ranking. This construction encodes listwise relevance preferences into term-level importance, enabling standard retrievers to approximate ranking-oriented relevance during retrieval.

The ranking-augmented query is then used to retrieve documents from the full corpus: $\mathcal{D}^{\text{rank}} = R(\tilde{q}^{\text{rank}}, \mathcal{C})$, allowing discriminative ranking signals extracted from a small candidate set to influence corpus-wide retrieval without training.

Instantiation Details of LRaR. In our implementation, we use a single LLM call to jointly perform query augmentation and listwise ranking,

$$\left(\tilde{q}^{\text{gen}},\, \pi,\, \mathbf{F}\right) = \mathcal{L}_{\text{LRaR}}\left(q,\, \mathcal{D}^{(0)}\right),$$

where $\tilde{q}^{\text{gen}}$ is the augmented query, π is a ranked permutation over $\mathcal{D}^{(0)}$, and $\mathbf{F}$ denotes auxiliary listwise feedback (e.g., a relevance cutoff or redundancy notes) extracted during comparative ranking. We design a unified prompt (Appendix A) that takes the query and top-k initial documents as input, and outputs both an augmented query and a ranked list with relevance indicators, enabling efficient inference with a single LLM call.

We instantiate the feedback $\mathbf{F}$ by marking a prefix of the ranked list as relevant, reformulating outputs such as "[1] > [3] > [5]" into "[1] > [3] | [5]", where documents before "|" are treated as relevant. To combine generative and discriminative signals, we fuse retrieval results from $\tilde{q}^{\text{gen}}$ and $\tilde{q}^{\text{rank}}$ via score summation with min–max normalization over the top-100 candidates. This fusion yields the final ranking used for evaluation.

4 Experiments

Evaluation. Following prior work (Sun et al., 2023; Liu et al., 2025), we evaluate LRaR on the TREC DL19 (Craswell et al., 2020), TREC DL20 (Craswell et al., 2021), and eight BEIR datasets (Thakur et al., 2021). NDCG@10 is used as the evaluation metric. We apply LRaR to augment BM25 retrieval.

Baselines. We compare LRaR against two families of baselines: query augmentation methods and listwise rerankers. In particular, we include the SOTA LameR (Shen et al., 2024) and RankGPT (Sun et al., 2023) method. For LRaR and baselines, we use the top-20 BM25-retrieved documents as the in-context candidate set. For a fair comparison, we implement both baselines using the same configuration as LRaR. For additional context, we also report results for several strong query augmentation methods, including HyDE (Gao et al., 2022), Query2doc (Wang et al., 2023b), MILL (Jia et al., 2024) and CSQE (Lei et al., 2024), which uses powerful models such as text-davinci-003-175B (Ouyang et al., 2022) and GPT-3.5-turbo.

Please refer to Appendix C for our detailed experimental implementations.

4.1 Main Results

Table 1 reports results on DL19/20 and BEIR. Our implementation of LameR consistently matches or outperforms prior query augmentation methods across datasets, providing a strong baseline. More importantly, compared to using reranking or query augmentation alone, LRaR yields consistent and substantial improvements across different models. For example, on BEIR Avg., LRaR improves performance from 51.27 to 53.38 (+2.11) when using the Qwen3-30B-A3B model. Overall, LRaR achieves the best results across all model variants on 8 out of 10 datasets, demonstrating its robustness and effectiveness across different retrieval settings.

In Table 2, we compare the efficiency–effectiveness trade-off of LRaR against conventional sliding-window reranking over the top-100 documents, reporting per-query LLM input/output tokens (on DL19) and retrieval effectiveness. RankGPT reranking with Qwen3 (4B) reaches 53.06 BEIR Avg. but consumes 16,914/810 input/output tokens. In contrast, LRaR achieves higher BEIR Avg. (54.15) with 7.6× fewer input tokens and 1.7× fewer output tokens. While top-100 reranking is slightly better on DL19/20, LRaR offers a markedly better cost–quality trade-off than naïvely applying LLM-based reranking over large candidate sets.

5 Analysis

In this section, we conduct analysis using the Qwen3-14B model.

Ablation Study. ablate the main components of LRaR and compare them to their closest baselines. As shown in Table 3, the reranking component performs on par with RankGPT reranking, and the query augmentation component is comparable to LameR augmentation. Combining the two signals in LRaR yields consistent improvements over using

Models	DL19	DL20	Covid	DBPedia	SciFact	NFCorpus	Signal	Robust04	Touche	News	BEIR Avg.
BM25	50.6	48.0	59.5	31.8	67.9	33.8	33.0	40.7	44.2	39.5	43.8
Reference Expansion Methods											
HyDE	61.3	57.9	59.3	36.8	69.1	-	-	-	-	44.0	-
Query2Doc	66.2	62.9	72.2	37.0	68.6	34.9	-	-	39.8	-	-
MILL	63.8	61.8	75.3	34.3	71.4	36.8	-	-	45.4	-	-
CSQE	67.3	66.2	74.2	40.3	69.6	-	-	-	-	48.7	-
RankGPT Reranking											
Qwen3 (4B)	60.8	60.3	69.9	37.3	72.0	36.1	33.5	48.7	39.5	43.9	47.6
Qwen3 (14B)	63.2	61.0	73.2	39.6	73.2	36.9	33.8	49.4	41.6	47.2	49.3
Qwen3 (30B-A3B)	63.5	61.8	71.5	39.4	73.6	36.4	35.5	48.5	44.1	47.3	49.5
LameR Query Augmentation											
Qwen3 (4B)	64.6	63.6	74.1	41.2	74.6	38.0	**36.9**	49.4	49.9	49.7	51.7
Qwen3 (14B)	66.3	63.4	75.0	41.2	74.4	38.2	34.5	50.2	49.3	50.0	51.6
Qwen3 (30B-A3B)	67.3	66.0	75.5	**41.6**	74.7	37.9	33.8	51.6	47.1	47.9	51.3
*LRaR **(Ours)***											
Qwen3 (4B)	66.2	**66.6**	**77.9**	40.5	76.1	**41.0**	35.5	**55.7**	49.7	52.4	**53.6**
Qwen3 (14B)	**67.7**	65.2	77.3	40.8	**76.6**	40.7	35.8	55.1	49.1	52.2	53.5
Qwen3 (30B-A3B)	67.6	66.3	75.0	41.5	75.3	39.9	35.6	54.9	**50.9**	**53.9**	53.4

Table 1: NDCG@10 on TREC DL19/20 and BEIR. The best results across all models are **bolded**.

	Token Consumption		Effectiveness		
	Input	Output	DL19	DL20	BEIR Avg.
BM25	-	-	50.6	48.0	43.8
Reference Fine-Tuned Rerankers on Top-100 Docs					
RankVicuna (7B)	-	-	67.7	66.0	49.0
RankZepyr (7B)	-	-	73.4	70.0	51.2
RankGPT Reranking on Top-100 Docs					
Qwen3 (4B)	16,914	810	69.1	68.2	53.1
LRaR on Top-20 Docs					
Qwen3 (4B)	2,212	473	66.2	66.6	54.2

Table 2: Efficiency-effectiveness trade-off of LRaR and reranking baseline. We report per-query LLM input/output tokens on DL19 and effectiveness (NDCG@10) on DL19/20 and BEIR.

Method	DL19	DL20	BEIR Avg.
RankGPT Reranking	63.2	61.0	49.3
LameR Expansion	66.3	63.4	51.6
LRaR	67.7	65.2	53.5
Reranking component	62.6	62.3	49.3
Query Aug. component	65.8	61.4	51.9
RRF rank fusion	63.9	61.0	51.5

Table 3: Ablation study on Qwen3-14B.

reranking or expansion alone. In contrast, directly fusing reranking and expansion with a Reciprocal Rank Fusion (RRF) scheme[1] fails to improve and even degrade performance, highlighting simple rank-level fusion is insufficient to propagate listwise ranking preferences into corpus-wide retrieval.

[1]Details of the rank fusion are in the Appendix B.

Method	DL19	DL20	BEIR Avg.
Original Ranking Order	62.6	62.3	49.3
Distillation Full Corpus	65.3	65.4	51.5
Distillation Rerank Candidates	62.2	60.3	49.6

Table 4: Analysis on reranking of LRaR on Qwen3-14B.

Analysis on Reranking Signal Distillation. We analyze the effect of our ranking-to-retrieval distillation in Table 4. Using distillation to construct a retrieval-compatible query and retrieve over the full corpus substantially improves effectiveness over applying the raw ranking order, indicating that distillation successfully propagates listwise preferences beyond the in-context set. Moreover, when we apply the distilled signal only to rerank the same candidate set originally used for listwise reranking, performance remains close to the original reranking results. Together, these results suggest that distillation largely preserves reranking precision while enabling corpus-wide generalization.

6 Conclusion

We presented LRaR, a zero-shot retrieval framework that augments LLM-based retrieval with listwise ranking signals extracted from a single LLM call. By distilling discriminative ranking preferences into retrieval-compatible queries, LRaR bridges generative augmentation and ranking-oriented relevance. Extensive experiments demonstrate that LRaR achieves strong, robust gains and a superior effectiveness-efficiency trade-off across diverse benchmarks and models.

Limitation

LRaR relies on the quality and stability of LLM-generated listwise rankings, which may vary across models, prompts, and domains. Errors or biases in the initial ranking can propagate to retrieval. Moreover, our current instantiation focuses on BM25-based retrieval and simple heuristic weighting, leaving more principled mappings and dense retrievers for future work.

References

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*.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2021. Overview of the trec 2020 deep learning track. *arXiv preprint arXiv:2102.07662*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

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. 2024. MILL: Mutual verification with large language models for zero-shot query expansion. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2498–2518, Mexico City, Mexico. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*, pages 6769–6781, Online.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626. Association for Computing Machinery.

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)*, pages 393–401. Association for Computational Linguistics.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2356–2362, Virtual Event, Canada. Association for Computing Machinery.

Wenhan Liu, Xinyu Ma, Yutao Zhu, Ziliang Zhao, Shuaiqiang Wang, Dawei Yin, and Zhicheng Dou. 2025. Sliding windows are not the end: Exploring full ranking with long-context large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 162–176. Association for Computational Linguistics.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2421–2425, New York, NY, USA. Association for Computing Machinery.

Iain Mackie, Shubham Chatterjee, and Jeffrey Dalton. 2023. Generative relevance feedback with large language models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 2026–2031, Taipei, Taiwan. Association for Computing Machinery.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online. Association for Computational Linguistics.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021. The expando-mono-duo design pattern for

text ranking with pretrained sequence-to-sequence models. *arXiv preprint arXiv:2101.05667*.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. RocketQAv2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2825–2835, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, and 1 others. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Xiaolong Huang, Binxing Jiao, Linjun Yang, and Daxin Jiang. 2023. LexMAE: Lexicon-bottlenecked pretraining for large-scale retrieval. In *The Eleventh International Conference on Learning Representations*.

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*, pages 15933–15946, Bangkok, Thailand. Association for Computational Linguistics.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2023a. SimLM: Pre-training with representation bottleneck for dense passage retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2244–2258, Toronto, Canada. Association for Computational Linguistics.

Liang Wang, Nan Yang, and Furu Wei. 2023b. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, Singapore. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2022. Adversarial retriever-ranker for dense text retrieval. In *International Conference on Learning Representations*.

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, Miami, Florida, USA. Association for Computational Linguistics.

A Prompt of LRaR

LRaR User Prompt

```
Search Query: {query}.
Passages:
[1] {passage 1}
[2] {passage 2}
...
[{num}] {passage {num}}

Task:
1) Write an expansion: a detailed answer-style passage addressing the query.
2) Rank ALL passages by relevance to the query (most relevant first), using passage identifiers.
Rules:
- You MUST include every passage identifier exactly once in the ranking.
- Put ALL and only highly relevant passages BEFORE '|'.
Ranking format example: [2] > [1]
Output format (exactly two lines):
Line 1: Expansion: <expansion text>
Line 2: Ranking: [i] > [j] > ... | [k] > ...
```

B Details on Rank Fusion

The rank fusion is performed with $\alpha = 60$, following common practice. We fuse top-20 reranking candidates and top-100 query augmentation retrieval results.

C Experimental Implementations.

We instantiate LRaR with models of varying sizes and architectures, including Qwen3-4B-Instruct-0725, Qwen3-14B, and the mixture-of-experts model Qwen3-30B A3B (Yang et al., 2025). We disable thinking mode for all models, as our preliminary experiments indicate that it incurs substantial overhead while providing only limited effectiveness gains for the models considered. For each query, we produce a single generation using sampling with temperature 1.0 via vLLM (Kwon et al., 2023). BM25 retrieval is performed with Pyserini (Lin et al., 2021) using default hyperparameters. We use the top-20 BM25-retrieved documents as the in-context candidate set. For a fair comparison, the baselines, LameR and RankGPT, use the same candidate set, models, and inference

configuration as LRaR. Finally, we compare the effectiveness-efficiency trade-off of LRaR against reranking applied to the BM25 top-100 candidates.