Iterative Reranking as a Compute-Scaling Method for LLM-based Rankers

Tamara Czinczoll[1]*[0009−0007−8224−1893],
Dong Liu[2][0000−0003−0394−1087],
Filippo Betello[3]*[0009−0006−0945−9688]

[1]Hasso Plattner Institute [2]Amazon, Luxembourg [3]Sapienza University of Rome

Abstract. E-commerce search faces challenges such as sparse data and poor generalization from issues like multi-attribute resolution, multi-hop reasoning, and implicit intent. We propose iterative reranking as a compute-scaling strategy for LLM-based rankers, repeatedly applying listwise rankers to refine results by exploiting LLM non-determinism. Evaluated on three open datasets with three open-source LLMs, the method trades increased computation for consistently improves performance, yielding strong nDCG@40 gains on DL19, FutureQueryEval, and difficult Amazon query types. These findings show that iterative reranking is an effective inference-time scaling approach for LLM rankers.

Keywords: Information Retrieval · Ranking · Reranking · LLM.

1 Introduction and Related Work

Search queries typically follow a long-tailed distribution, where infrequent queries suffer from limited data availability and poor generalization. To address this, we investigate iteratively applying listwise LLM-based rankers to difficult queries, a process where the output ranking from one pass serves as the input order for the next. Unlike pointwise or pairwise approaches, listwise ranking enables crucial inter-document judgments [14], yet it faces challenges regarding computational costs, limited context windows, and positional bias. Recent efforts to scale inference-compute for better ranking have primarily explored generating extended reasoning traces, [18] and [16]. Other work has focused on efficiently fitting candidates into context windows, [12], [7]. Our approach, by contrast, seeks to translate additional compute into quality by allowing the model to iteratively refine its own permutations.

While iterative or multi-pass strategies have been touched upon in other work, they have largely remained on the fringes of research, usually treated as auxiliary experiments rather than primary subjects of study. For instance, RankVicuna [8] and RankZephyr [9] include iterative reranking only as an ablation experiment, without exploring generalization beyond their proprietary

[*] Work done during an internship at Amazon.

Query Difficulty	Description	Example
Vague	Implicit exploratory intent Ambiguous queries	Student Gaming Laptop
Multi-Attribute	Queries with many specifications and/or constraints	wireless noise-cancelling headset under $200 with at least 30 hours of battery life
Comparative	Comparative or preference-based queries	Cheaper alternative to AirPods
Negation	Queries with negations	Non-toxic nail polish for toddlers
Natural Language	Full-sentence descriptions	I need a compact blender that's easy to clean and good for making smoothies
Others	Other aspects that make answering the query difficult, such as code switching or use of abbreviations	Queries with temporal aspects (seasonal, time-sensitive). Queries requiring context of previous search history

Table 1. Overview of the query difficulty types used to annotate the Amazon dataset.

finetuned models. Similarly, Liu et al. [6] utilize multiple passes solely to generate labels for finetuning, offering no analysis of the quality changes per pass. The closest prior attempt, Qin et al. [10], applies a pairwise ranker over multiple passes; however, their investigation is relegated to an appendix and lacks an in-depth exploration of design decisions or generalizability.

In this work, we move beyond peripheral observations to provide a systematic study of iterative listwise reranking. Drawing inspiration from human cognitive processes, where complex tasks are often solved through step-by-step refinement [13], [5], [2], we provide the first systematic investigation of this method. We evaluate the approach across two passage datasets and one product dataset, using LLMs to annotate the latter for difficulty sources. Our analysis reveals that while iterative, listwise reranking generally improves passage ranking, performance gains in product ranking are highly dependent on the specific query difficulty type.

2 Iterative Reranking

We propose iterative reranking where at step t of T total reranking steps, given a search query q that is part of an instruction prompt p^t, and a list l_n^t of n ranked items at step t, our goal is to refine the current ranking list into a new ranking at step $t+1$

$$l_n^{t+1} = LLM(l_n^t | p^t) \tag{1}$$

so that given an evaluation function $eval$ that measures the quality of the ranking $eval(l_n^T) > eval(l_n^0)$. The evaluation function $eval$ is usually nDCG, MAP or MRR at different cutoff ranges, when ranking is based on relevance. To ensure that we can rerank a given list of documents without cutting off documents due to the LLM's finite context length, we follow [9] and employ a sliding window approach. The window starts ranking the bottom M documents and slides upwards with a stride of S, until it reaches the top. One full sliding window pass constitutes one ranking iteration. At each iteration t, the LLM uses the previous iteration's ranking l_n^t to produce an updated ranking l_n^{t+1}.

2.1 Experimental Setup

Datasets: We evaluate our approach on three datasets: (i) a curated dataset from Amazon Shopping Queries [11], using a targeted 1000-query-sample. This subset was extracted from the 22,458 English test queries which consists of real user search data and product rankings annotated by crowd workers. This subset is intentionally hard, with 80% of the queries populated with difficult query types. We annotated queries with the difficulty types from Table 1 by the majority vote of three LLMs. The dataset is structured to ensure a minimum representation of each difficult query type. The remaining 20% are non-hard queries. This focus on difficult queries provides the necessary context for the performance analysis per query difficulty type discussed in Section 3; (ii) TREC DL19 [4], a well-established passage ranking benchmark with 43 annotated test queries and 9,260 documents from MSMARCO; and (iii) FutureQueryEval [1] contains 147 manually created queries about post-April 2025 events across seven topics, to test ranking performance on data beyond current LLM training cutoffs.
Models: We evaluate three large language models: RankZephyr, fine-tuned on ranking data; Qwen3-8B [17] and Gemma3-4B[15], two dense LLMs without dedicated ranking finetuning.
Evaluation: The two metrics, nDCG@5 and nDCG@40 [3, 19], are chosen to evaluate ranking quality at different levels of user exposure, with nDCG@5 focusing on highly visible, immediate results and nDCG@40 on a broader range.

3 Results

Table 2 presents results for RankZephyr, Qwen and Gemma on the three datasets. Iterative reranking always improves performance by iteration five on the FutureQueryEval dataset for both nDCG@5 and @40, and it always improves nDCG@40 on DL19, whereas it is more mixed for nDCG@5. On the Amazon dataset, only RankZephyr showcases better performance. Whether the the prompt instructs to rank from scratch (default) or refine an existing list ('Improve') has a small, model-dependent impact. On the Amazon dataset the other models gradually degrade with subsequent iterations. We further investigate this phenomenon by presenting nDCG@5 performance per query difficulty type for the first and ninth iterations in Figure 1.

The analysis shows that improvement depends on the query's difficulty. Qwen improves rankings for queries of all classes except multi-attribute ones. RankZephyr slightly improves all classes but the 'Other' category. For Gemma the impact of iterative reranking is more mixed, possibly due to its smaller size of only

	nDCG@5					nDCG@40				
Iteration	1	2	3	4	5	1	2	3	4	5
Amazon Subset										
- RankZephyr - Default	75.95	76.64	76.61	**76.80**	76.74↑	87.79	**87.94**	87.83	87.92	87.88↑
- RankZephyr - “Improve”	75.95	**76.47**	76.14	76.41	76.18↑	87.79	**87.90**	87.68	87.82	87.67↓
- Qwen3-8B - Default	**82.05**	80.72	80.00	80.23	79.89↓	**90.55**	89.77	89.34	89.47	89.32↓
- Qwen3-8B - “Improve”	**82.13**	80.34	79.51	79.77	79.69↓	**90.59**	89.54	89.01	89.20	89.08↓
- Gemma3-4B - Default	**76.58**	75.31	74.44	74.35	73.78↓	**86.94**	86.89	86.39	86.48	85.95↓
- Gemma3-4B - “Improve”	**76.58**	74.37	71.85	72.80	72.21↓	**86.94**	86.32	84.92	85.47	84.87↓
TREC DL19										
- RankZephyr - Default	67.71	**67.76**	66.96	67.03	67.03↓	53.35	55.68	55.90	**56.51**	56.15↑
- RankZephyr - “Improve”	67.61	67.65	**67.78**	67.52	67.52↓	53.35	55.75	56.35	56.53	**56.60**↑
- Qwen3-8B - Default	65.54	65.91	67.16	66.47	**67.34**↑	52.11	54.82	56.06	56.17	**56.35**↑
- Qwen3-8B - “Improve”	65.12	65.34	65.68	**66.59**	65.68↑	51.89	54.46	55.80	**56.26**	56.10↑
- Gemma3-4B - Default	61.53	62.23	60.25	**62.62**	60.73↓	49.57	51.31	51.78	**52.33**	52.27↑
- Gemma3-4B - “Improve”	61.80	61.68	61.02	60.71	**62.18**↑	49.64	51.11	51.77	52.18	**52.78**↑
FutureQueryEval										
- RankZephyr - Default	62.44	62.76	62.85	62.86	**62.88**↑	65.55	65.97	66.12	66.14	**66.17**↑
- RankZephyr - “Improve”	62.44	62.54	**62.83**	62.82	62.82↑	65.55	65.90	66.06	66.06	**66.10**↑
- Qwen3-8B - Default	61.11	62.48	62.86	**63.16**	62.86↑	64.46	65.44	65.56	**66.11**	65.77↑
- Qwen3-8B - “Improve”	61.07	62.08	62.40	**62.61**	62.31↑	64.38	65.54	65.63	**65.79**	65.54↑
- Gemma3-4B - Default	54.33	56.13	**56.99**	55.48	56.74↑	59.85	61.62	**61.98**	60.92	61.81↑
- Gemma3-4B - “Improve”	54.56	56.07	**57.52**	56.92	57.49↑	59.95	61.27	**62.09**	61.73	62.07↑

Table 2. nDCG@k in %, $k \in \{5, 40\}$ when iteratively reranking for five iterations with different ranking prompts. Unlike the default prompt, the “Improve” prompt asks to improve the given ranking. Results are aggregated over five seeds. Best performance per model and metric over all iterations is shown in **bold**. The arrows at iteration five indicate whether performance improved or degraded compared to the initial ranking.

four billion parameters compared to the others' seven and eight billion. Surprisingly, iterative reranking has a strong impact on the performance of comparative queries, i.e., those comparing (parts of) the product against another. Here, the two LLMs more than double nDCG@5. While this effect needs to be further studied due to the limited number of comparative queries in our data subset, we hypothesize that the strong listwise, inter-product dependencies that need to be captured to answer these queries particularly benefit from iterative reranking.

4 Conclusion

We investigated iterative reranking as a compute-scaling strategy to refine listwise rankings. The method yielded consistent gains on passage ranking benchmarks (DL19, FutureQueryEval). However, performance on e-commerce product ranking was mixed and highly model-dependent: while RankZephyr showed overall improvement, other models degraded over multiple iterations, with success heavily relying on specific query difficulties. Iterative reranking can be of particular use if no larger suitable model is available. Future work should explore optimal routing strategies, develop adaptive stopping criteria, and more precisely

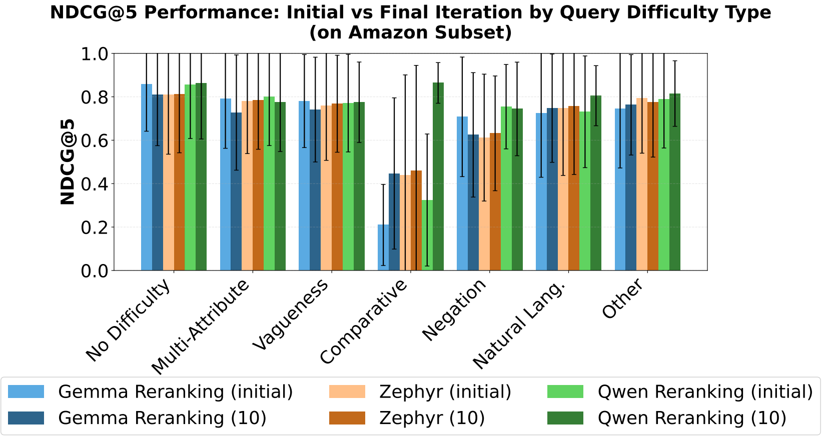

Fig. 1. NDCG@5 performance by query difficulty type on the Amazon Shopping Queries subset, comparing the initial and tenth reranking iterations.

evaluate the trade-offs between ranking quality improvements and the associated computational overhead across different query types and domains.

CV of the Presenters: *Tamara Czinczoll* is a PhD student at the Hasso Plattner Institute under the supervision of Prof. Gerard De Melo. She specializes in LLM pretraining, focusing on the foundational architectures that drive modern AI. *Dong Liu* is an Applied Scientist at Amazon. He holds a PhD from the KTH Royal Institute of Technology. His research interests include IR, probabilistic graphical models, and Bayesian inference, with a focus on applying these methods to large-scale industrial environments. *Filippo Betello* is a PhD student at Sapienza University of Rome, supervised by Prof. Fabrizio Silvestri. His research bridges Recommender Systems, IR, and LLMs, with a specific emphasis on computational efficiency and minimizing the carbon footprint.
Company: Amazon operates one of the world's largest and most complex e-commerce search engines, serving millions of customers and indexing billions of products daily. Dealing with massive scale and distinct data heterogeneity, Amazon focuses on developing high-performance, low-latency systems that solve semantic matching challenges to seamlessly connect customers with the products they need across the global retail ecosystem.

Acknowledgments. We would like to thank Amine Moukadiri, Maryna Beliuga and Paul Missault for sharing their expertise and providing helpful feedback.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Abdallah, A., Piryani, B., Mozafari, J., Ali, M., Jatowt, A.: How Good are LLM-based Rerankers? An Empirical Analysis of State-of-the-Art Reranking Models. In: Christodoulopoulos, C., Chakraborty, T., Rose, C., Peng, V. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2025. pp. 5693–5709. Association for Computational Linguistics, Suzhou, China (Nov 2025). https://doi.org/10.18653/v1/2025.findings-emnlp.305, https://aclanthology.org/2025.findings-emnlp.305/
2. Adams, R., Atman, C.: Cognitive processes in iterative design behavior. In: FIE'99 Frontiers in Education. 29th Annual Frontiers in Education Conference. Designing the Future of Science and Engineering Education. Conference Proceedings (IEEE Cat. No.99CH37011. vol. 1, pp. 11A6/13–11A6/18 vol.1 (Nov 1999). https://doi.org/10.1109/FIE.1999.839114, https://ieeexplore.ieee.org/abstract/document/839114, iSSN: 0190-5848
3. Bellogín, A., Castells, P., Cantador, I.: Statistical biases in Information Retrieval metrics for recommender systems. Information Retrieval Journal **20**(6), 606–634 (Dec 2017). https://doi.org/10.1007/s10791-017-9312-z, https://doi.org/10.1007/s10791-017-9312-z
4. Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Voorhees, E.M.: OVERVIEW OF THE TREC 2019 DEEP LEARNING TRACK
5. Lewis, D.G.R., Gorson, J., Maliakal, L.V., Carlson, S.E., Gerber, E.M., Riesbeck, C.K., Easterday, M.W.: Planning to Iterate: Supporting Iterative Practices for Real-world Ill- structured Problem-solving (2018)
6. Liu, W., Ma, X., Zhu, Y., Zhao, Z., Wang, S., Yin, D., Dou, Z.: Sliding Windows Are Not the End: Exploring Full Ranking with Long-Context Large Language Models. In: Che, W., Nabende, J., Shutova, E., Pilehvar, M.T. (eds.) Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 162–176. Association for Computational Linguistics, Vienna, Austria (Jul 2025). https://doi.org/10.18653/v1/2025.acl-long.8, https://aclanthology.org/2025.acl-long.8/
7. Parry, A., MacAvaney, S., Ganguly, D.: Top-Down Partitioning for Efficient List-Wise Ranking (May 2024). https://doi.org/10.48550/arXiv.2405.14589, http://arxiv.org/abs/2405.14589, arXiv:2405.14589 [cs]
8. Pradeep, R., Sharifymoghaddam, S., Lin, J.: RankVicuna: Zero-Shot Listwise Document Reranking with Open-Source Large Language Models (Sep 2023), https://arxiv.org/abs/2309.15088v1
9. Pradeep, R., Sharifymoghaddam, S., Lin, J.: RankZephyr: Effective and Robust Zero-Shot Listwise Reranking is a Breeze! (Dec 2023), https://arxiv.org/abs/2312.02724v1
10. Qin, Z., Jagerman, R., Hui, K., Zhuang, H., Wu, J., Yan, L., Shen, J., Liu, T., Liu, J., Metzler, D., Wang, X., Bendersky, M.: Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting. In: Findings of the Association for Computational Linguistics: NAACL 2024. pp. 1504–1518. Association for Computational Linguistics, Mexico City, Mexico (2024). https://doi.org/10.18653/v1/2024.findings-naacl.97, https://aclanthology.org/2024.findings-naacl.97
11. Reddy, C.K., Màrquez, L., Valero, F., Rao, N., Zaragoza, H., Bandyopadhyay, S., Biswas, A., Xing, A., Subbian, K.: Shopping Queries Dataset: A Large-Scale ESCI Benchmark for Improving Product Search (Jun 2022).

https://doi.org/10.48550/arXiv.2206.06588, http://arxiv.org/abs/2206.06588, arXiv:2206.06588 [cs]

12. Ren, R., Wang, Y., Zhou, K., Zhao, W.X., Wang, W., Liu, J., Wen, J.R., Chua, T.S.: Self-Calibrated Listwise Reranking with Large Language Models. In: Proceedings of the ACM on Web Conference 2025. pp. 3692–3701. WWW '25, Association for Computing Machinery, New York, NY, USA (Apr 2025). https://doi.org/10.1145/3696410.3714658, https://dl.acm.org/doi/10.1145/3696410.3714658
13. Schön, D.A.: The reflective practitioner: how professionals think in action. Basic Books, New York (1983)
14. Sun, W., Yan, L., Ma, X., Wang, S., Ren, P., Chen, Z., Yin, D., Ren, Z.: 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. pp. 14918–14937. Association for Computational Linguistics, Singapore (2023). https://doi.org/10.18653/v1/2023.emnlp-main.923, https://aclanthology.org/2023.emnlp-main.923
15. Team, G., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M., Rouillard, L., Mesnard, T., Cideron, G., Grill, J.b., Ramos, S., Yvinec, E., Casbon, M., Pot, E., Penchev, I., Liu, G., Visin, F., Kenealy, K., Beyer, L., Zhai, X., Tsitsulin, A., Busa-Fekete, R., Feng, A., Sachdeva, N., Coleman, B., Gao, Y., Mustafa, B., Barr, I., Parisotto, E., Tian, D., Eyal, M., Cherry, C., Peter, J.T., Sinopalnikov, D., Bhupatiraju, S., Agarwal, R., Kazemi, M., Malkin, D., Kumar, R., Vilar, D., Brusilovsky, I., Luo, J., Steiner, A., Friesen, A., Sharma, A., Sharma, A., Gilady, A.M., Goedeckemeyer, A., Saade, A., Feng, A., Kolesnikov, A., Bendebury, A., Abdagic, A., Vadi, A., György, A., Pinto, A.S., Das, A., Bapna, A., Miech, A., Yang, A., Paterson, A., Shenoy, A., Chakrabarti, A., Piot, B., Wu, B., Shahriari, B., Petrini, B., Chen, C., Lan, C.L., Choquette-Choo, C.A., Carey, C.J., Brick, C., Deutsch, D., Eisenbud, D., Cattle, D., Cheng, D., Paparas, D., Sreepathihalli, D.S., Reid, D., Tran, D., Zelle, D., Noland, E., Huizenga, E., Kharitonov, E., Liu, F., Amirkhanyan, G., Cameron, G., Hashemi, H., Klimczak-Plucińska, H., Singh, H., Mehta, H., Lehri, H.T., Hazimeh, H., Ballantyne, I., Szpektor, I., Nardini, I., Pouget-Abadie, J., Chan, J., Stanton, J., Wieting, J., Lai, J., Orbay, J., Fernandez, J., Newlan, J., Ji, J.y., Singh, J., Black, K., Yu, K., Hui, K., Vodrahalli, K., Greff, K., Qiu, L., Valentine, M., Coelho, M., Ritter, M., Hoffman, M., Watson, M., Chaturvedi, M., Moynihan, M., Ma, M., Babar, N., Noy, N., Byrd, N., Roy, N., Momchev, N., Chauhan, N., Sachdeva, N., Bunyan, O., Botarda, P., Caron, P., Rubenstein, P.K., Culliton, P., Schmid, P., Sessa, P.G., Xu, P., Stanczyk, P., Tafti, P., Shivanna, R., Wu, R., Pan, R., Rokni, R., Willoughby, R., Vallu, R., Mullins, R., Jerome, S., Smoot, S., Girgin, S., Iqbal, S., Reddy, S., Sheth, S., Põder, S., Bhatnagar, S., Panyam, S.R., Eiger, S., Zhang, S., Liu, T., Yacovone, T., Liechty, T., Kalra, U., Evci, U., Misra, V., Roseberry, V., Feinberg, V., Kolesnikov, V., Han, W., Kwon, W., Chen, X., Chow, Y., Zhu, Y., Wei, Z., Egyed, Z., Cotruta, V., Giang, M., Kirk, P., Rao, A., Black, K., Babar, N., Lo, J., Moreira, E., Martins, L.G., Sanseviero, O., Gonzalez, L., Gleicher, Z., Warkentin, T., Mirrokni, V., Senter, E., Collins, E., Barral, J., Ghahramani, Z., Hadsell, R., Matias, Y., Sculley, D., Petrov, S., Fiedel, N., Shazeer, N., Vinyals, O., Dean, J., Hassabis, D., Kavukcuoglu, K., Farabet, C., Buchatskaya, E., Alayrac, J.B., Anil, R., Dmitry, Lepikhin, Borgeaud, S., Bachem, O., Joulin, A., Andreev, A., Hardin, C., Dadashi, R., Hussenot, L.: Gemma

3 Technical Report (Mar 2025). https://doi.org/10.48550/arXiv.2503.19786, http://arxiv.org/abs/2503.19786, arXiv:2503.19786 [cs]

16. Weller, O., Ricci, K., Yang, E., Yates, A., Lawrie, D., Durme, B.V.: Rank1: Test-Time Compute for Reranking in Information Retrieval (Feb 2025). https://doi.org/10.48550/arXiv.2502.18418, http://arxiv.org/abs/2502.18418, arXiv:2502.18418 [cs] version: 1
17. Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., Qiu, Z.: Qwen3 Technical Report (May 2025). https://doi.org/10.48550/arXiv.2505.09388, http://arxiv.org/abs/2505.09388, arXiv:2505.09388 [cs]
18. Yang, E., Yates, A., Ricci, K., Weller, O., Chari, V., Durme, B.V., Lawrie, D.: Rank-K: Test-Time Reasoning for Listwise Reranking (May 2025). https://doi.org/10.48550/arXiv.2505.14432, http://arxiv.org/abs/2505.14432, arXiv:2505.14432 [cs] version: 1
19. Yu, H.T.: Optimize What You Evaluate With: A Simple Yet Effective Framework For Direct Optimization Of IR Metrics (Aug 2020). https://doi.org/10.48550/arXiv.2008.13373, http://arxiv.org/abs/2008.13373, arXiv:2008.13373 [cs]