

Rank-R1: Enhancing Reasoning in LLM-based Document Rerankers via Reinforcement Learning

Anonymous ACL submission

Abstract

Existing document reranking methods based on large language models (LLMs) typically rely on prompting or fine-tuning LLMs to order or label candidate documents according to their relevance to a query. In this paper, we introduce Rank-R1, an LLM-based reranker that performs reasoning over both the user query and candidate documents before performing the ranking task, by training with reinforcement learning along with only a small set of relevance labels (without any reasoning supervision) to enhance the reasoning ability of LLM-based rerankers. Our hypothesis is that adding reasoning capabilities to the rerankers can improve their relevance assessment and ranking capabilities. Our experiments on the TREC DL and BRIGHT datasets show that Rank-R1 is highly effective, especially for complex queries. In particular, we find that Rank-R1 achieves effectiveness on in-domain datasets at par with that of supervised fine-tuning methods. In addition, the model largely outperforms zero-shot and supervised fine-tuning when applied to out-of-domain datasets featuring complex queries, especially when a 14B-size model is used.

1 Introduction

Large language models (LLMs) have shown strong performance in document ranking tasks (Ma et al., 2023; Sun et al., 2023; Zhuang et al., 2023, 2024b,a; Sun et al., 2024; Xu, 2024). Zero-shot prompting methods such as RankGPT, rerank documents by directly prompting LLMs to generate reordered document lists (Sun et al., 2023). However, these methods rely on the assumption that the LLM follows instructions well. Moreover, being zero-shot, they do not leverage available human-annotated relevance data for further improvement.

In contrast, methods like RankLlama fine-tune LLMs using human relevance judgments, assigning scores to each query-document pair (Ma et al., 2024). While effective, these approaches do not

explicitly model reasoning processes mainly due to the lack of high-quality reasoning data for supervised fine-tuning. In practice, user relevance judgments come in the form of selecting the most relevant document from a set, but achieving high-quality rankings requires strong reasoning capabilities to interpret complex relevance relationships.

Recent advancements in reinforcement learning (RL) for LLMs, such as DeepSeek-R1 (Guo et al., 2025), have shown success in improving LLM reasoning in tasks like mathematical question-answering (Shao et al., 2024). These recent innovations suggest that rule-based reward scoring alone can improve an LLM’s ability to reason and explain.

Building on this insight, we ask whether reinforcement learning in the style of Deepseek-R1 can improve reasoning in document reranking. Specifically, we apply Group Relative Policy Optimization (GRPO) (Shao et al., 2024) to train an LLM-based reranker. Given a user query and a list of retrieved candidate documents, the reranker generates reasoning steps before selecting the most relevant document. The reward signal is determined only by whether the model eventually identifies the most relevant document among the candidates correctly.

We train our model, Rank-R1, on the MS MARCO passage ranking dataset and evaluate it on TREC DL19 and DL20 (in-domain datasets). Our results show that RL-based training is at par with supervised fine-tuning on in-domain data.

Additionally, we evaluate Rank-R1 on the BRIGHT dataset (SU et al., 2025), which requires complex query reasoning and relevance understanding, and is out-of-domain with respect to the data used to train the model. Rank-R1, trained with retrieval reinforcement learning, outperforms both zero-shot prompting and supervised fine-tuning on this out-of-domain dataset. Notably, our 14B model surpasses the much larger (zeroshot) GPT-4 in reranking performance on the BRIGHT dataset.

2 Method

2.1 LLM Reranking

Our LLM reranker follows the Setwise prompting approach proposed by Zhuang et al. (2024b). This method takes a query and a set of candidate documents as input to the LLM and prompts the LLM to select the most relevant document among the candidates based on relevance to the query. Then, the heapsort algorithm is used to build a heap tree over all the candidate documents from the first-stage retriever, and the documents are reranked via the “heapify” operations with the Setwise prompt.

However, the original Setwise ranking approach does not encourage the LLM to reason about the relevance between the query and the documents. Instead, it directly asks for the most relevant one. To unlock the reranker’s reasoning ability, we modify the original Setwise prompt by adding a reasoning instruction, as shown in Figure 1. Specifically, we adapt the system prompt from the DeepSeek-R1-Zero method to the Setwise prompt. This modification encourages LLMs to reason first before providing an answer—in our case, predicting the label of the most relevant candidate document to the query. We refer to the Setwise method using this modified prompt as *Rank-R1*.

SYSTEM:

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within `<think>` `</think>` and `<answer>` `</answer>` tags, respectively, i.e., `<think>` reasoning process here `</think>` `<answer>` answer here `</answer>`.

USER:

Given the query: "{query}", which of the following documents is most relevant?

[1] {document1}

[2] {document2}

....

[20] {document20}

After completing the reasoning process, please provide only the label of the most relevant document to the query, enclosed in square brackets, within the answer tags. For example, if the third document is the most relevant, the answer should be: `<think>` reasoning process here `</think>` `<answer>`[3]`</answer>`.

Figure 1: Prompt used for Rank-R1.

2.2 Reinforcement Learning

Although any modern instruction-tuned LLM, when coupled with our *Rank-R1* prompt, may exhibit strong zeroshot reasoning ability, their rea-

soning process could still be suboptimal for the Setwise ranking method. This is because the LLMs might not have been fine-tuned on similar instructional data. On the other hand, gathering human-annotated reasoning data for large-scale supervised fine-tuning of *Rank-R1* could be both costly and infeasible. To address these challenges, we employ the GRPO RL algorithm (Shao et al., 2024) to enhance the reasoning process of the *Rank-R1* reranker. GRPO is a reinforcement learning algorithm that fine-tunes large language models by sampling multiple responses per prompt, evaluating them with a reward model, and updating the policy to favor higher-reward responses without requiring a separate value function, thereby enhancing efficiency and stability. We provided the details of the algorithm in Appendix A.

The Setwise prompts and rewards for our GRPO training come from training data. For constructing such RL training data, we use the MS-MARCO training data¹ provided by the Tevatron IR toolkit (Gao et al., 2023), which includes training queries, human-labeled relevant documents, and BM25-retrieved top-100 documents. For each training query, we sample 19 documents from the set retrieved by BM25, along with one labeled relevant document, to form the *Rank-R1* prompt.

We design the reward function for RL as follows: a reward of one is granted if and only if the LLM generations match the reasoning and answering format (i.e., the generated tokens fill in the `<think>` `</think>` `<answer>` `</answer>` spans) and the answer correctly matches the label of the ground-truth relevant document. Otherwise, a reward of zero is provided. Our straightforward rule-based reward mechanism does not impose constraints on the reasoning process but instead encourages the model to generate the correct format and answer following the reasoning. Most importantly, this RL training does not require human-annotated reasoning data.

There are two main reasons why we selected the Setwise ranking approach as the backend of our Rank-R1: First, the nature of Setwise prompting allows us to use a simple rule-based reward function, as only the most relevant document label needs to be predicted, enabling a straightforward match with the answer span. Secondly, it is convenient for us to fairly compare the RL-trained Rank-R1 with the supervise fine-tuned Setwise reranker, which sim-

¹<https://huggingface.co/datasets/Tevatron/msmarco-passage>

Model	Training	DL19	DL20
BM25	zeroshot	50.6	48.0
RankZephyr-7B	GPT4-distil.	73.9	70.6
RankGPT	Zeroshot	75.6	70.6
Setwise-3B	Zeroshot	37.1	31.7
Setwise-3B	SFT	73.4	67.2
Rank-R1-3B	Zeroshot	60.5	53.8
Rank-R1-3B	GRPO	71.3	66.8
Setwise-7B	Zeroshot	67.5	63.6
Setwise-7B	SFT	73.8	69.2
Rank-R1-7B	Zeroshot	71.2	66.2
Rank-R1-7B	GRPO	72.7	68.5
Setwise-14B	Zeroshot	67.7	64.8
Setwise-14B	SFT	72.9	68.9
Rank-R1-14B	Zeroshot	67.9	65.2
Rank-R1-14B	GRPO	71.4	69.1

Table 1: TREC DL19 and DL20 nDCG@10 results. SFT=supervised fine tuned. GRPO trained on only 18% of 400k data used to train SFT.

ply removes the reasoning process from the prompt and is directly trained to predict the ground-truth label using the same training data. Other LLM-based ranking methods, such as Listwise prompting, are harder to fairly compare in this setting, as a ground-truth ranking is usually not available in the training data (MSMARCO only has one judged relevant document per query on average).

3 Experimental Settings

Datasets. We explore to dataset settings to evaluate the effectiveness of Rank-R1: an in-domain setting, where we use the TREC-DL19 and DL20 datasets (Craswell et al., 2020, 2021), and an out-of-domain setting, where we use the BRIGHT benchmark datasets (SU et al., 2025). The DL19 and DL20 are in-domain because they are based on the MSMARCO passage ranking dataset – the same dataset used in the training of our rerankers; unlike MSMARCO though they contain deep assessments for each query (around 210 assessments per query on average). The BRIGHT benchmark datasets encompass domains, including biology, code, and math, and require intensive reasoning to rank relevant documents.

Initial Retrieval. For all methods we consider in our experiments, the initial retrieval is performed using the *pyserini* implementation of BM25 (Lin et al., 2021). Reranking approaches are then provided the top 100 documents retrieved by BM25 to rerank.

Rank-R1 Settings. We compare two settings of Rank-R1: (i) a Zeroshot setting, where only the Setwise-based prompt, improved by the presence

of the reasoning instruction is used, with no training performed, and (ii) a GRPO setting, where we employ the same prompt as in the Zeroshot setting, but we also train the model according to the GRPO RL method. For GRPO, we set $|G| = 8$, that is, for each training query we generate 8 answers with the old policy $\pi_{\theta_{old}}$; as training dataset we use the training part of MS MARCO. For all Rank-R1 experiments, due to the limited computational resources, we only train on approximately 18% of the full MS-MARCO dataset, requiring roughly three (3B and 7B models) to five (14B model) days of training on four H100 GPUs. The details of GRPO training hyperparameters are provided in Appendix C.

Comparison Methods. To understand the effect of the reasoning prompt and the GRPO training on the Setwise approach, we compare Rank-R1 against the original Setwise method (using prompt as listed in Appendix B), either used in a Zeroshot manner, or trained with the standard supervised fine-tuning (SFT) on MS MARCO data (400k training datapoints circa). The details of SFT training hyperparameters are provided in Appendix C.

To further contextualise the effectiveness of Rank-R1, we also consider the effectiveness of the current state-of-the-art Listwise reranker, RankZephyr (Pradeep et al., 2023), which was trained with data generated from GPT-4. We use the model checkpoint² released by the author and run it ourselves to ensure the exact same settings. We also report the effectiveness of RankGPT (Sun et al., 2023), a zero-shot listwise reranked where GPT-4 is used as backbone. Note that for the BRIGHT dataset, RankGPT was applied to a different implementation of BM25 from the one we used: the one used by RankGPT has a higher average nDCG@10 than our BM25.

Backbone LLMs. For Setwise and Rank-R1, we explore base LLMs using instruction-tuned Qwen2.5 series models (Qwen et al., 2025) ranging from 3 billion to 14 billion parameters. RankZephyr is based on the Zephyr 7B backbone (Tunstall et al., 2023); RankGPT is based on OpenAI’s GPT-4 model (OpenAI et al., 2024).

4 Results

We present both in-domain effectiveness results and out-of-domain zero-shot results in this section.

²https://huggingface.co/castorini/rank_zephyr_7b_v1_full

Model	Training	Bio.	Earth.	Econ.	Psy.	Rob.	Stack.	Sus.	Pony	Leet.	AoPS	TheoT.	TheoQ.	Avg.
BM25	zeroshot	.182	.279	.164	.134	.109	.163	.161	.043	.247	.065	.021	.073	.137
RankZephyr-7b	GPT4-distill	.219	.237	.144	.103	.076	.137	.166	.065	.247	.068	.020	.073	.130
RankGPT4*	Zeroshot	.338	.342	.167	.270	.223	.277	.111	.156	.034	.012	.086	.002	.170
Setwise-3B	Zeroshot	.143	.175	.120	.102	.077	.079	.154	.053	.154	.017	.042	.021	.095
Setwise-3B	SFT	.220	.188	.104	.115	.091	.058	.167	.057	.099	.040	.034	.038	.101
Rank-R1-3B	Zeroshot	.137	.173	.119	.152	.100	.066	.178	.037	.077	.040	.060	.025	.097
Rank-R1-3B	GRPO	.184	.171	.137	.169	.090	.100	.165	.047	.111	.035	.059	.032	.108
Setwise-7B	Zeroshot	.236	.223	.161	.171	.149	.092	.183	.063	.149	.041	.104	.056	.136
Setwise-7B	SFT	.287	.301	.141	.239	.189	.137	.196	.071	.207	.070	.082	.082	.167
Rank-R1-7B	Zeroshot	.268	.248	.179	.221	.174	.103	.211	.044	.156	.033	.104	.059	.150
Rank-R1-7B	GRPO	.260	.285	.172	.242	.191	.104	.242	.043	.198	.043	.109	.083	.164
Setwise-14B	Zeroshot	.295	.322	.205	.248	.189	.147	.236	.087	.187	.080	.093	.076	.180
Setwise-14B	SFT	.220	.293	.154	.230	.201	.157	.203	.062	.194	.095	.099	.097	.167
Rank-R1-14B	Zeroshot	.301	.366	.221	.246	.217	.154	.250	.090	.170	.091	.116	.092	.193
Rank-R1-14B	GRPO	.312	.385	.212	.264	.226	.189	.275	.092	.202	.097	.119	.092	.205

Table 2: BRIGHT nDCG@10 results. All methods rerank BM25 top-100 documents (First line). *: Results copied from the paper which uses a different BM25 ranking system (has a higher average nDCG@10 than our BM25).

In Appendix E.1, we also provide an analysis of reward trends and response lengths during GRPO training, along with a case study.

4.1 In-domain effectiveness

In Table 1, we present the effectiveness of Rank-R1 variants on the TREC-DL19 and DL20 passage ranking datasets. We start by comparing Setwise and Rank-R1 under the zero-shot setting. The results suggest that incorporating the reasoning process into the Setwise method improves zero-shot ranking effectiveness. The improvements are particularly large on the 3B size model.

Next we consider the effects of training with GRPO. With GRPO training, Rank-R1 effectiveness increases, indicating that reasoning and answer generation (i.e. ranking) are enhanced by RL training. This improvement makes Rank-R1 comparable to the Setwise SFT trained on the full dataset and brings it closer to the effectiveness of the state-of-the-art RankZephyr.

Since the results in Table 1 for Rank-R1 trained with GRPO are obtained using only 18% of the MSMARCO training data (while SFT used all available training data), we explore the effect of training data quantity in Appendix D. There, we provide comparative results for both GRPO and SFT trained with varying amounts of data.

4.2 Reasoning intensive out-of-domain effectiveness

Next we consider results from our out-of-domain experiments, reported in Table 2.

We observe that the SOTA RankZephyr reranker, which does not incorporate reasoning, does not provide better rankings than BM25 in most datasets from the BRIGHT benchmark – effectively fail-

ing at the reranking task. This suggests that the BRIGHT benchmark poses a challenge for current SOTA LLM rerankers.

On the other hand, Rank-R1 trained with GRPO outperforms or is on par with both zero-shot and Setwise SFT models in most cases. Notably, when using the 14B model, Setwise SFT effectiveness plateaued and even performed worse than its zero-shot counterparts, suggesting that the large model trained on the MSMARCO could not generalize to the BRIGHT using the standard Setwise approach. However, Rank-R1 based on the 14B model achieves the largest performance gain over Setwise SFT and even surpasses the GPT-4-based Listwise reranker baseline (RankGPT4). These results highlight that the reasoning process can help the model generalize to different domains and that strong reasoning abilities, along with larger model sizes, are crucial for LLM-based rerankers to be effective in reasoning-intensive ranking tasks.

5 Conclusion

In this paper, we present Rank-R1, a Setwise reranker trained using the GRPO reinforcement learning strategy to enhance its reasoning capabilities. Our training approach does not require human-annotated reasoning data; instead, it relies only on document relevance assessments. While RL achieves performance comparable to supervised fine-tuning on in-domain tasks, Rank-R1 significantly outperforms state-of-the-art non-reasoning Listwise rerankers on reasoning-intensive benchmarks. These results highlight the value of integrating reasoning into document reranking, with potential benefits for both effectiveness and the explainability of LLM-based systems, as well as improved result presentation in search engines.

6 Limitation

In this work, we used MS MARCO Passage ranking as the only training dataset to study the in-domain and zero-shot effectiveness of our model on MS MARCO and BRIGHT benchmark. While this setup enables a controlled analysis, extending Rank-R1 to more diverse collections—such as BGE (Xiao et al., 2023) or recently released reasoning-oriented datasets like ReasonIR (Shao et al., 2025) has the potential to further enhance its generalization and reasoning capabilities. We leave this as an avenue for future work. Additionally, our experiments constrained the input context length for each document to 1024 tokens. However, real-world documents often exceed this limit. Lastly, our study focused exclusively on English retrieval tasks. Adapting Rank-R1 to multilingual or cross-lingual retrieval settings warrants future investigation.

References

- Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. [Overview of the trec 2020 deep learning track](#). *Preprint*, arXiv:2102.07662.
- Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. [Overview of the trec 2019 deep learning track](#). *Preprint*, arXiv:2003.07820.
- Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. [Tevatron: An efficient and flexible toolkit for neural retrieval](#). In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 3120–3124, New York, NY, USA. Association for Computing Machinery.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *arXiv:2501.12948*.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. [Search-r1: Training llms to reason and leverage search engines with reinforcement learning](#). *Preprint*, arXiv:2503.09516.
- 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 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, pages 2356–2362.
- 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.
- Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. [Zero-shot listwise document reranking with a large language model](#). *Preprint*, arXiv:2305.02156.
- Tong Niu, Shafiq Joty, Ye Liu, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. [Judgerank: Leveraging large language models for reasoning-intensive reranking](#). *Preprint*, arXiv:2411.00142.
- OpenAI et al. 2024. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.
- Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. [Rankzephyr: Effective and robust zero-shot listwise reranking is a breeze!](#) *Preprint*, arXiv:2312.02724.
- Qwen et al. 2025. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muen-nighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen tau Yih, Pang Wei Koh, and Luke Zettlemoyer. 2025. [Reasonir: Training retrievers for reasoning tasks](#). *Preprint*, arXiv:2504.20595.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *Preprint*, arXiv:2402.03300.
- Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025. [R1-searcher: Incentivizing the search capability in llms via reinforcement learning](#). *Preprint*, arXiv:2503.05592.
- Hongjin SU, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han yu Wang, Liu Haisu, Quan Shi, Zachary S Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan O Arik, Danqi Chen, and Tao Yu. 2025. [BRIGHT: A realistic and challenging benchmark for reasoning-intensive retrieval](#). In *The Thirteenth International Conference on Learning Representations*.

Shuoqi Sun, Shengyao Zhuang, Shuai Wang, and Guido Zuccon. 2024. An investigation of prompt variations for zero-shot llm-based rankers. *arXiv preprint arXiv:2406.14117*.

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, Singapore. Association for Computational Linguistics.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Cl  mentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. *Zephyr: Direct distillation of lm alignment*. *Preprint*, arXiv:2310.16944.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Galou  dec. 2020. Trl: Transformer reinforcement learning. <https://github.com/huggingface/trl>.

Orion Weller, Kathryn Ricci, Eugene Yang, Andrew Yates, Dawn Lawrie, and Benjamin Van Durme. 2025. *Rank1: Test-time compute for reranking in information retrieval*. *Preprint*, arXiv:2502.18418.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. *C-pack: Packaged resources to advance general chinese embedding*. *Preprint*, arXiv:2309.07597.

Zhichao Xu. 2024. *Rankmamba: Benchmarking mamba’s document ranking performance in the era of transformers*. *Preprint*, arXiv:2403.18276.

Honglei Zhuang, Zhen Qin, Kai Hui, Junru Wu, Le Yan, Xuanhui Wang, and Michael Bendersky. 2024a. *Beyond yes and no: Improving zero-shot LLM rankers via scoring fine-grained relevance labels*. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 358–370, Mexico City, Mexico. Association for Computational Linguistics.

Shengyao Zhuang, Bing Liu, Bevan Koopman, and Guido Zuccon. 2023. *Open-source large language models are strong zero-shot query likelihood models for document ranking*. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8807–8817, Singapore. Association for Computational Linguistics.

Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2024b. *A setwise approach for effective and highly efficient zero-shot ranking with large language models*. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*

’24, page 38–47, New York, NY, USA. Association for Computing Machinery.

A GPRO Algorithm

The GRPO algorithm optimizes the following objective:

$$\begin{aligned} \mathcal{J}_{GRPO}(\theta) = & \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{|G|} \sim \pi_{\theta_{old}}(O|q)] \\ & \frac{1}{|G|} \sum_{i=1}^{|G|} \left(\min \left(\frac{\pi_{\theta}(o_i|q)}{\pi_{\theta_{old}}(o_i|q)} A_i, \right. \right. \\ & \left. \left. \text{clip} \left(\frac{\pi_{\theta}(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}, 1 - \epsilon, 1 + \epsilon \right) A_i \right) \right. \\ & \left. - \beta \mathbb{D}_{KL}(\pi_{\theta} || \pi_{ref}) \right) \end{aligned} \quad (1)$$

where Q is the Setwise ranking prompt in the training data, G is a group of generated data points sampled from an old policy $\pi_{\theta_{old}}$ (in our case, the initial LLM) given a sampled prompt, and π_{θ} is the LLM ranker we are optimizing. The clip operation is used to cap the ratio of the new and old policies. In our experiment, we only use the data samples from $\pi_{\theta_{old}}$ to update π_{θ} once, meaning $\pi_{\theta_{old}} = \pi_{\theta}$, so Eq 1 reduces to:

$$\begin{aligned} \mathcal{J}_{GRPO}(\theta) = & \\ & \frac{1}{|G|} \sum_{i=1}^{|G|} \left(\frac{\pi_{\theta}(o_i|q)}{\pi_{\theta_{old}}(o_i|q)} A_i, -\beta \mathbb{D}_{KL}(\pi_{\theta} || \pi_{ref}) \right) \end{aligned} \quad (2)$$

where \mathbb{D}_{KL} is the KL loss that penalizes how far the new policy differs from a reference policy π_{ref} , which in our case is the original instruction-tuned LLM.

The generated data points, in our case, represent the reasoning process. We use the generated relevant document label to calculate the advantage A as follows:

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \dots, r_G\})}{\text{std}(\{r_1, r_2, \dots, r_G\})} \quad (3)$$

where r is the reward given by the training data, in our case a reward of one is granted if and only if the LLM generations match the reasoning and answering format, and the answer correctly matches the label of the ground-truth relevant document. Otherwise, a reward of zero is provided.

Overall, the GRPO training optimizes the LLM to generate tokens that maximize the rewards.

Parameter	GRPO	SFT
learning_rate	$1e-5$	$1e-5$
batch_size	64	64
optimizer	AdamW	AdamW
lora_rank	16	16
max_prompt_length	4096	4096
max_completion_length	2048	na
group_size	8	na

Table 3: Hyper-parameters for training.

B Prompt for Setwise Reranker

SYSTEM:

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant provides the user with the answer enclosed within `<answer>` `</answer>` tags, i.e., `<answer>` answer here `</answer>`.

USER:

Given the query: "{query}", which of the following documents is most relevant?

[1] {document1}

[2] {document2}

....

[20] {document20}

Please provide only the label of the most relevant document to the query, enclosed in square brackets, within the answer tags. For example, if the third document is the most relevant, the answer should be: `<answer>`[3]`</answer>`.

We use the above prompt for both zero-shot and supervised fine-tuning of Setwise rerankers. The only difference from the prompt used for *Rank-R1* is that the reasoning instructions are removed.

C Training hyper-parameters

We use the TRL (von Werra et al., 2020) library for both GPRP and SFT training. Both methods are trained with LoRA adapter (Hu et al., 2022). For SFT, we compute the cross-entropy loss only on the answer span and ignore the loss on the prompt tokens. The hyperparameters are listed in Table 3, while other parameters follow the default settings of the TRL trainer.

D Effect of Quantity of Training Data

The results in Table 1 for Rank-R1 trained with GRPO are obtained when using only 18% of the MSMARCO training data (while SFT used all available training data). To explore whether longer training could further improve effectiveness, we continued training the 3B and 7B Rank-R1 models

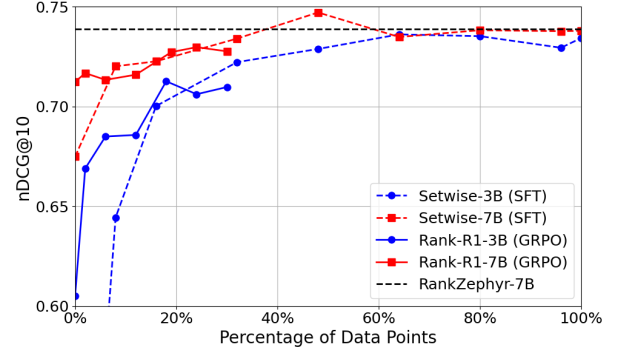


Figure 2: Data efficiency comparison between Setwise SFT and Rank-R1.

for an additional two days and evaluated checkpoints saved during training. We report the results in Figure 2. In the figure, we also include results obtained when using SFT on incremental parts of the training data.

From the figure, we observe that Rank-R1 requires significantly less data than Setwise SFT to achieve the same level of performance at early training stage – however this data efficiency effect vanishes early on during the training phase. Passed 5-7% of training data, in fact, the two training approaches tend to track each other. SFT has a clear advantage over GRPO in that it is by far less computationally expensive. On the other hand, GRPO adds new features to the reranker, introducing the ability to perform reasoning.

E Analysis

E.1 Reward score v.s. Response length

In Figure 3, we present the received reward values and model completion lengths logged during training for Rank-R1, across different model sizes. Rewards consistently increase throughout training, with smaller models showing a higher rate of increase, while larger models start with a higher initial reward.

Regarding completion length, larger models tend to generate longer responses; however, we do not observe a noticeable increase in length as training proceeds. This observation differs from the findings for DeepSeek-R1 (Guo et al., 2025). This may be attributed to two factors. First, we initialize RL training from an instruction-tuned model rather than a base model, meaning the instruction model already follows a reasonable reasoning process. Second, the MSMARCO passage ranking dataset is relatively simple compared to tasks like

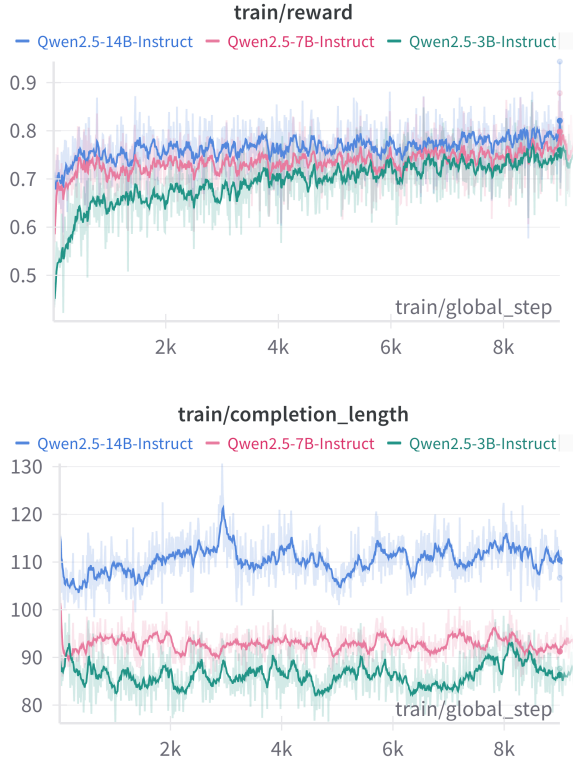


Figure 3: Rewards (top) and model completion length (bottom) obtained during GRPO training.

math or coding, where a longer reasoning process is more essential. Thus, extensive reasoning may not be necessary for achieving high effectiveness in this task.

E.2 Case study

In Figure 4, we provide an example of Rank-R1’s generation. We compare the outputs of the Zero-shot model and the model after GPRO training. Both models successfully follow the instruction by providing a reasoning process within the `<think>` span and predicting a relevant document label in the correct format. However, the Zero-shot model tends to merely describe what each document mentions and ultimately makes an incorrect prediction. In contrast, the GPRO-trained model focuses on the most relevant documents, compares them, and correctly selects the best one. In addition, we argue that Rank-R1’s transparent reasoning process makes its predictions more explainable, which could be particularly important in domains such as medical document ranking.

Query: common minerals list

Zero-shot Rank-R1 Completion:

`<think>`The query is "common minerals list". Document [3] talks about mica being a common rock-forming mineral. Document [19] mentions that chemical sedimentary rocks are formed through precipitation of minerals. However, it does not provide a comprehensive list of common minerals. Documents [1], [2], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], and [20] do not mention anything related to common minerals.`</think>`
`<answer>`[3]`</answer>`

GRPO-Trained Rank-R1 Completion:

`<think>`The query is "common minerals list". Document [3] talks about mica, which is a common rock-forming mineral. Document [4] also mentions common rock-forming minerals. Both documents seem relevant to the query. However, document [4] provides a list of common rock-forming minerals, which directly answers the query. `</think>`
`<answer>`[4]`</answer>`

Figure 4: Example of completion. The Rank-R1 used in this example is based on Qwen2.5-3B-Instruct. The GRPO-Trained Rank-R1 gives the correct answer.

F Related Works

Recent efforts have been made to introduce reasoning capabilities into information retrieval models. For example, Search-R1 (Jin et al., 2025) and R1-Searcher (Song et al., 2025) both leverage reinforcement learning to enhance the search capabilities of the retriever in the RAG pipeline. In contrast, our work focuses on the re-ranking task, using reasoning to improve the model’s ability to select the relevant documents.

To introduce reasoning into the re-ranking task, JudgeRank (Niu et al., 2024) proposes a multi-step prompting strategy to analyze query and document content before re-ranking. However, this is a zero-shot method that relies on the LLM’s inherent knowledge and instruction-following ability. Concurrent work, Rank1 (Weller et al., 2025), is the most similar to ours, applying knowledge distillation from DeepSeek-R1 to a reranker. In contrast, our work focuses on directly improving the model’s reasoning ability for the ranking task through reinforcement learning.