# *InsertRank*: LLMs can reason over BM25 scores to Improve Listwise Reranking


## Abstract

Large Language Models (LLMs) have demonstrated significant strides across various information retrieval tasks, particularly as rerankers, owing to their strong generalization and knowledge-transfer capabilities acquired from extensive pretraining. In parallel, the rise of LLM-based chat interfaces has raised user expectations, encouraging users to pose more complex queries that necessitate retrieval by "reasoning" over documents rather than through simple keyword matching or semantic similarity. While some recent efforts have exploited reasoning abilities of LLMs for reranking such queries, considerable potential for improvement remains. In that regards, we introduce ***InsertRank***, an LLM-based reranker that leverages lexical signals like BM25 scores during reranking to further improve retrieval performance. ***InsertRank*** demonstrates improved retrieval effectiveness on – BRIGHT, a reasoning benchmark spanning 12 diverse domains, and R2MED, a specialized medical reasoning retrieval benchmark spanning 8 different tasks. We conduct an exhaustive evaluation and several ablation studies and demonstrate that ***InsertRank*** consistently improves retrieval effectiveness across multiple families of LLMs, including GPT, Gemini, and Deepseek models. With Deepseek-R1, ***InsertRank*** achieves a score of 37.5 on the BRIGHT benchmark. and 51.1 on the R2MED benchmark, surpassing previous methods. In addition, we additionally demonstrate the effectiveness of ***InsertRank*** on standard benchmarks like TREC DL 19, 20 and TREC HARD, further demonstrating the robustness of this method. In addition, we also demonstrate the effectiveness of our method with BERT based retriever scores, thus illustrating how including feedback from the first stage retriever can be helpful to guide a listwise LLM reranker.



## CCS Concepts

• **Information systems** → **Retrieval models and ranking**; • **Computing methodologies** → *Natural language processing*.

## Keywords

reranking, reasoning, large language models, retrieval

**ACM Reference Format:**

. 2018. ***InsertRank***: LLMs can reason over BM25 scores to Improve Listwise Reranking. In *Proceedings of — (InsertRank).* ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX





*InsertRank, Retrieval*





ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX


## 1 Introduction

Large language models (LLMs) have shown immense success across various tasks in natural processing and information retrieval. They have been successfully applied to reformulate queries [6, 22] and documents [16], rerank documents [13, 17] and products [5], as well as train embeddings for dense indices [19]. They are pretrained on huge corpora, and often excel at a variety of search tasks, often without the need for additional fine-tuning.

In the context of reranking, they have been often used through 3 paradigms – pointwise, pairwise, and listwise. The most common reranking models are cross encoders, which have a BERT backbone [18], ColBERT rerankers [19], which work on late interaction, and LLM-based rerankers. In many prior works, BM25 is utilized as a common first-stage retrieval system, from which the top-k documents are fed to the reranker.

While there is prior work leveraging LLMs in all three settings (point, pair, and listwise), we focus on the listwise setting in this particular work. A listwise approach is advantageous as it consumes lesser number of tokens than the corresponding aggregated pointwise approaches, and allows the LLM to reason over all the documents at once. Moreover, with more and more LLMs having the ability to handle very long contexts, listwise reranking is becoming viable for LLM-based document reranking. In this work, we explore the use of LLMs on re-ranking passages in the context of complex reasoning-centric queries. To that end, we utilize two benchmarks - BRIGHT [21] and R2MED [10] - a medical reasoning retrieval benchmark. Particularly, we experiment with the injection of BM25 scores into the prompt and demonstrate it as a useful signal for reasoning-centric LLM reranking. Injecting BM25 scores into the ranking input along with query and document has proven to be effective for BERT based cross encoders [2]. In this work, we demonstrate it is a useful signal to augment the reasoning capabilities of LLMs in the reranking setting.

In this work, we ask the following research question – ***Can incorporating lexical signals, such as retrieval scores, serve as effective clues for rerankers to improve retrieval effectiveness in reasoning tasks ?***

Specifically, our work contributes the following:

(1) We introduce ***InsertRank*** – a simple listwise reranking method that exploits BM25 retrieval scores to improve retrieval over reasoning queries
(2) We evaluate our method across multiple open and closed LLMs to demonstrate the effectiveness across two reasoning centric retrieval benchmarks - BRIGHT and R2MED
(3) We also conduct ablations and analyses along the following dimensions:
   (a) Given the long context of the reranking inputs in the listwise setting, and tendencies of LLMs to favor context that is at the beginning and the end, we perform ablation experiments by shuffling the document order with and without BM25.

(b) We experiment with scale, normalization to examine their effects on the reasoning abilities of LLMs in the reranking context.

*While many studies have focused on enhancing reasoning through fine-tuning and reinforcement learning methods, which rely heavily on labeled data, to the best of our knowledge, ours is the first work demonstrating improved retrieval effectiveness by integrating retrieval scores within a zero-shot setting using a listwise generative reranker.*

## 2 Related work

We now discuss related work to place our contributions in context.

### 2.1 Retrieval for Complex Queries

Reasoning-centric queries are often much more difficult and nuanced than those in traditional document retrieval, where keyword matching or semantic matching suffice. With LLMs becoming increasingly more powerful at reasoning and understanding, they become crucial for improving ranking and retrieval effectiveness for complex reasoning queries [23] [21]. BRIGHT [21] is a challenging benchmark of $\tilde{1}300$ queries across 11 domains and $\tilde{1}$M documents. Similarly, R2MED [10] is a challenging benchmark of 876 queries across 8 tasks in the medical domain, which focuses on reasoning-centric retrieval. On BRIGHT, Su et al. [21] have observed significant gains with query reformulation using GPT-4, Gemini, and other LLMs with a BM25 backbone.

There has also been growing research around training retrievers and rankers for reasoning-centric information retrieval. Shao et al. [20] finetune a LLama-8B model for complex reasoning queries. They also develop a synthetic data generation pipeline that produces complex hard negatives for finetuning a dense retrieval model. Weller et al. [23] leverage reasoning traces that are collected from Deepseek-R1 on the MSMARCO dataset and fine-tune small language models of varying sizes to achieve significant results on the BRIGHT benchmark. [25] leverage the GRPO technique to finetune language models of varying sizes (3B to 14B) for listwise reasoning centric reranking. Yang et al. [24] uses a listwise reranker finetuned on QwQ-32B and leverage a sliding window approach in a listwise setting to reduce the number of LLM calls compared to the pointwise setting. Niu et al. [15] leverages innovative prompting strategies with GPT-4 to score queries and documents in a pointwise setting. While many works have focused on improving reasoning using finetuning and RL methods, to the best of our knowledge, ours is the first work to incorporate BM25 scores into the LLM prompt for a *listwise zero shot setting.*

### 2.2 LLM Based Reranking

In the context of reranking, there are mainly three salient paradigms - pointwise, pairwise and listwise, with a fourth one namely setwise that has been recently introduced.

- **Pointwise**: Produce a score $s_j$ for each pair $(q_i, D_j)$ where $q_i$ is the ith query and $d_j$ is the jth document in the evaluation.

$$(q_i, D_j) \rightarrow \mathbf{M} \rightarrow s_j \tag{1}$$

- **Pairwise**: Produce a preference score $s_j$ for each triple of the form $(q_i, D_j, D_k)$ ; the goal is to maximize the number of instances where $(q_i, D_j, D_k) > 0$ when $D_j$ is more relevant than $D_k$ in the ground truth where $(q_i, D_j, D_k) > 0$ indicates $D_j$ is more relevant than $D_k$.

$$(q, D_j, D_k) \rightarrow \mathbf{M} \rightarrow s_j \tag{2}$$

- **Listwise**: The goal in a listwise setting is to consider a query $q_i$ and a list of documents $D_1 ... D_n$ and produce a ranked list that takes in all the documents from the retriever at once.

$$(q, D_1, \ldots, D_n) \rightarrow \mathbf{M} \rightarrow r_1\, r_2\, \ldots\, r_n \tag{3}$$

where $r_1, r_2, ... r_n$ are the ranked list of documents or their identifiers.

### 2.3 Leveraging numerical information in language models

There have been several prior works studying how both BERT based (encoder) and decoder only LLMs understand numerics. [12] provide a comprehensive survey of mathematical LLMs, covering CoT, tool use, instruction tuning, etc. Similarly, [1] provides an overview of LLM abilities in problem solving, math reasoning, geometry and so on. Askari et al. [2] fine-tuned a BERT based cross encoder reranker by injecting BM25 scores along with the document tokens and found improvements over a pointwise cross encoder setup. However, it is unclear whether BM25 scores can boost recent LLM capabilities under more realistic settings – namely without fine-tuning, and in listwise reranking. Our work demonstrates the **effectiveness of injecting BM25 scores in a zero shot listwise setting with no finetuning**.

## 3 Proposed Method

We now describe our proposed method. ***InsertRank*** involves injecting the retriever's BM25 score into the listwise reranking setting.

$$(q, D_1, b_1 \ldots, D_n, b_n) \rightarrow \mathbf{M} \rightarrow r_1, r_2 \ldots r_l \tag{4}$$

Here, $q$ is the query, $D_1, D_2, ... D_n$ is the list of documents passed to the reranker, $b_1, b_2, ... b_n$ is the BM25 scores associated with each document, and $r_1, r_2, ... r_n$ is the reranked list of document identifiers.

In addition, in our experiments with BRIGHT and R2MED, the documents are passed in decreasing order of their BM25 scores, i.e $b_1 > b_2 > ... b_n$

For the BRIGHT benchmark, inspired by the Rank-1 paper, we leverage their queries augmented by GPT-4 chain of thought (CoT) as it is reported to give the best NDCG@10 scores on BM25. Similarly, for the R2MED benchmark, we leverage the HyDE query reformulation [7] mentioned in their work, where a hypothetical document is leveraged as a query reformulator. as it is reported to give the best NDCG@10 scores in their BM25 first stage retrieval setting. In a listwise setting with BM25 score injection, the queries and documents along with the scores are passed as follows,

> <instructions> You are also given the BM25 scores from a lexical retrieval sytem. <query> <doc_1, BM25 score: s_1, doc_2, BM25 score: s_2, .... doc_n, BM25 score: b_n>

Other than the ablation setting in Table 4, *documents are ordered by decreasing order of BM25 scores.*

In a reasoning setting, LLMs are often known to have issues like hallucinations [8], incorrect reasoning [9], brittleness with

**Table 1: Performance in BRIGHT benchmark (P - Pointwise, L - Listwise, only retrieval if neither P nor L mentioned)**

| | Bio | Earth | Econ | Psy | Robot | Stack | Sust | LC | Pony | AoPS | TheoQ | TheoT | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BM25 | .192 | .271 | .149 | .125 | .135 | .165 | .152 | .244 | .079 | .060 | .130 | .069 | .148 |
| BM25 on GPT-4o CoT | .536 | .536 | .243 | .386 | .188 | .227 | .259 | .193 | .177 | .039 | .189 | .202 | .265 |
| Gemini 2.0 flash | .556 | .520 | .286 | .488 | .300 | .315 | .415 | .203 | .303 | .052 | .225 | .348 | .334 |
| ***+InsertRank*** | **.594** | **.556** | **.309** | .512 | .307 | .305 | .391 | **.224** | .265 | **.070** | **.240** | **.371** | **.345** |
| Gemini 2.5 flash | .540 | .543 | .254 | .403 | .231 | .306 | .302 | .224 | .195 | .053 | .224 | .241 | .293 |
| ***+InsertRank*** | **.596** | **.564** | **.298** | **.496** | **.298** | **.326** | **.391** | .207 | **.253** | .053 | .237 | **.377** | **.341** |
| GPT-4o | .598 | .556 | .299 | .530 | .346 | .328 | .426 | .229 | .311 | .077 | .308 | .409 | .368 |
| ***+InsertRank*** | **.626** | **.578** | **.326** | .505 | .317 | **.349** | .422 | **.264** | .226 | **.091** | **.328** | **.426** | **.371** |
| Deepseek-r1 | .530 | .507 | .325 | **.530** | .349 | **.365** | **.472** | **.224** | **.243** | .089 | .343 | .487 | .372 |
| ***+InsertRank*** | **.582** | **.527** | **.347** | .517 | **.362** | .348 | .428 | .220 | .229 | **.102** | **.351** | **.489** | **.375** |

**Table 2: Performance in R2MED benchmark**

| | Bioinfo | Biology | IIYi-Clin | Med-Sci | MedQA-Diag | MedXQA | PMC-Clin | PMC-Treat | Avg |
|---|---|---|---|---|---|---|---|---|---|
| BM25 (HyDE) | .430 | .573 | .104 | .412 | .496 | .265 | .328 | .406 | .377 |
| Gemini 2.0 flash (L) | **.588** | .534 | **.156** | .559 | **.566** | **.371** | **.477** | .593 | .480 |
| ***+InsertRank*** | .581 | **.603** | .145 | **.566** | .562 | .351 | .450 | **.614** | **.484** |
| Gemini 2.5 flash (L) | .660 | .577 | **.220** | .580 | .578 | .427 | **.535** | .624 | .525 |
| ***+InsertRank*** | **.684** | **.630** | .185 | **.587** | **.589** | **.467** | .524 | **.628** | **.537** |
| GPT-4o (L) | **.627** | .567 | .186 | **.594** | .594 | .377 | .423 | **.637** | .501 |
| ***+InsertRank*** | .622 | **.602** | .186 | .590 | **.596** | **.382** | **.429** | .626 | **.504** |
| Deepseek-r1 (L) | .615 | .559 | **.201** | **.612** | .581 | **.394** | .437 | **.664** | .508 |
| ***+InsertRank*** | **.639** | **.576** | .196 | .600 | **.605** | .376 | **.442** | .652 | **.511** |

respect to changing numbers and names [14]. Overthinking is also established as a common issue in reasoning models - a tendency where LLMs tend to produce very verbose reasoning chains for simpler problems leading to issues like concept drift [3] [4]. By providing a critical lexical relevance signal like BM25 scores, the goal is to ground the reasoning and prevent the model from running into issues like overthinking and concept drift and ground the reasoning process with respect to the first stage retriever.

By injecting the BM25 scores in the LLM reranking step, we provide a low cost solution for improving reasoning centric LLM reranking. Our solution produces consistent improvements across two reasoning centric retrieval benchmarks with no extra cost of finetuning and negligible additional token costs.

For all our experiments, we utilize the official repository of BRIGHT and R2MED. For the LLM implementations of Gemini-2.0-flash, GPT-4o and Deepseek-R1, we use their respective official APIs.

## 4 Results

The results of our experiment are as shown in Table 1 and 2. The top performing setting scores an average of **37.5** on the BRIGHT benchmark. We observe consistent gains by injecting BM25 scores into the prompt on multiple LLM families - Gemini, GPT-4 and Deepseek. The results show consistent improvement over a vanilla LLM listwise reranking with just queries and documents. By injecting BM25 scores into the prompt, we show gains of 3.2% on Gemini 2.0 flash, 16.3% on Gemini 2.5 flash, 0.8% on GPT-4o and Deepseek-r1 compared to just using the raw queries and documents. While we are able to use full length of the documents in the Gemini models, due to context length limitations, we use only the first 1800 tokens for the GPT-4o and Deepseek series of models.

We observe similar gains in the R2MED benchmark, with BM25 injection consistently surpassing the ranking quality on average compared to the vanilla listwise setting which takes just the documents into the prompt. The gains demonstrated are 0.8% for Gemini 2.0 flash, 2.2% in Gemini 2.5 flash and 0.5% gains in GPT-4o and Deepseek family of models.

## 5 Ablations

In this section, we analyze the effectiveness of our proposed method in the context of 1) normalization 2) shuffling input documents

### 5.1 Scaling and normalization of BM25

We additionally also examine the effect of different scales and how LLMs perceive them. Since BM25 scores are normally not restricted to a particular range, we perform an experiment with normalized BM25 scores. Similar to the previous experiments, we do two settings – one with BM25 injected and one without. Finally, we also examine the effect of scaling wherein normalized scores are scaled from 0-100.

The ablation results are reported for both R2MED and BRIGHT on the Gemini 2.0 flash model. As evidenced in 3, R2MED shows around 0.4% decrease when BM25 scores are normalized from 0-1

**Table 3: Effect of normalized BM25 scores**

| | BRIGHT | R2MED |
|---|---|---|
| **Setting** | **Avg** | **Avg** |
| Raw BM25 scores | **.345** | .484 |
| 0-1 scale | .340 | .482 |
| 0-100 scale | .342 | **.488** |

and 0.8% increase when BM25 scores are normalized from 0-100. While there is a small decrease in the 0-1 normalization setting, the 0-100 normalization, shows a marginal improvement.

Similarly for BRIGHT, we observe similar marginal performance gains when using a 0-100 normalization and very slight decrease when scores are normalized from 0-1. The results listed in table 3 indicate a 0.58% decrease when normalizing from 0-1 and 0.5% increase when normalizing using 0-100. With NDCG@10 scores across normalization also beating the vanilla listwise setting , **the results demonstrate the robustness of *InsertRank* to normalization and scaling.**

## 5.2 Shuffling order of documents

LLMs are well known to prefer documents at the beginning and end of context [11]. In order to validate the effectiveness of the proposed approach we shuffle the document tuples, where each tuple is of the form $D, B$ where $D$ is the document and $B$ is the BM25 score associated with the document. Similar to the previous experiments, we perform two settings, one with the BM25 scores injected and one without it. Unlike, normalization, we observe divergent results - while BRIGHT benchmark demonstrates robustness in the BM25 injection and shows gains, R2MED on the other hand, shows a consistent decrease when the documents are shuffled. Similar to the previous ablation, we report results for both BRIGHT and R2MED on Gemini 2.0 flash. As evidenced in 4, BRIGHT in the shuffled setting with BM25 injection demonstrates a 9.4% increase relative to the vanilla setting. However, there is a 1.1 points absolute decrease compared to the original setting, where documents are passed in decreasing order of BM25 scores. This establishes that listwise reranking methods in general are very sensitive to initial ordering for reasoning centric retrieval/reranking.

**Table 4: Effect of shuffling on R2MED**

| | BRIGHT | R2MED |
|---|---|---|
| **Setting** | **Avg** | **Avg** |
| Shuffled | .285 | **.488** |
| Shuffled w/ BM25 | .322 | .445 |

## 5.3 Performance on standard retrieval benchmarks

In addition to reasoning centric retrieval benchmarks like BRIGHT and R2MED, we also demonstrate ***InsertRank***'s effectiveness in standard retrieval datasets like TREC DL 19, 20 and TREC DL HARD. We retrieve top 100 documents using a first stage retriever and feed them into the LLM for reranking. As seen in 5, we see consistent improvements (+0.95%, 1.3%, and 6.0% NDCG@10 over vanilla LLM reranking on DL'19, DL'20, and DL-Hard respectively.) relative to a vanilla LLM reranker, when feedback from first stage retrieval (in this case BM25) is provided.

| Method | DL'19 | DL'20 | DL-Hard |
|---|---|---|---|
| BM25 | 0.5058 | **0.4803** | 0.2849 |
| Gemini 2.0 flash | 0.5366 | 0.4448 | 0.2768 |
| + ***InsertRank*** | **0.5417** | 0.4506 | **0.2934** |

**Table 5: NDCG@10 on TREC Deep Learning benchmarks.**

## 5.4 Performance with BERT based retrievers

In addition to BM25, we also demonstrate the effectiveness of ***InsertRank*** using 1) BERT based retriever and injecting them into the reranker 2) using a combination of BERT based retrieval and injecting BERT and BM25 scores into the reranking process. Across the TREC Deep Learning benchmarks, InsertRank yields consistent gains over vanilla LLM reranking: on DL'19, the dense variant improves NDCG@10 by +4.94%; on DL'20, by +7.31%; and on DL-Hard, by +33.82% (dense). Similar to the previous setting, we take the first 100 documents from a BERT based retriever (msmarco-v1-passage.bge-base-en-v1.5) and feed it into the LLM reranking pipeline, along with their BERT similarity scores

| Method | DL'19 | DL'20 | DL-Hard |
|---|---|---|---|
| Gemini 2.0 Flash | 0.6375 | 0.6444 | 0.2812 |
| + ***InsertRank*** (dense) | 0.6690 | 0.6915 | 0.3763 |

**Table 6: NDCG@10 of LLM reranking pipelines with and without *InsertRank* on TREC Deep Learning benchmarks.**

## 6 Conclusion and Future Work

In this paper, we present ***InsertRank***, a simple test time strategy that incorporates feedback from the first stage retriever, improving LLM reranking for reasoning centric retrieval tasks. By incorporating BM25 based lexical signals, we are able to consistently improve retrieval effectiveness across a wide variety of tasks on multiple reasoning centric retrieval benchmarks.

Our work also opens up new avenues for research as to what other low-cost relevance signals and additional meta-data could reasoning behavior take advantage from to improve reranking further? LLMs' reasoning abilities at inference time enable improved document ranking performance by supporting zero-shot reasoning over documents with their BM25 scores, a phenomenon not previously observed in non-reasoning zero-shot scenarios. This indicates that the reasoning capacity of LLMs offers a promising avenue for exploiting possibly other metadata, potentially enriching document representation and retrieval.

Moreover, our work also encourages other reranking applications to look for additional metadata, such as in distillation, which depend on exploiting reasoning traces of larger models.

## References


[1] Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large Language Models for Mathematical Reasoning: Progresses and Challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, Neele Falk, Sara Papi, and Mike Zhang (Eds.). Association for Computational Linguistics, St. Julian's, Malta, 225–237. https://aclanthology.org/2024.eacl-srw.17/

[2] Arian Askari, Amin Abolghasemi, Gabriella Pasi, Wessel Kraaij, and Suzan Verberne. 2023. Injecting the BM25 Score as Text Improves BERT-Based Re-rankers. arXiv:2301.09728 [cs.IR] https://arxiv.org/abs/2301.09728

[3] Hieu Tran Bao, Nguyen Cong Dat, Nguyen Duc Anh, and Hoang Thanh-Tung. 2025. Learning to Stop Overthinking at Test Time. arXiv:2502.10954 [cs.CV] https://arxiv.org/abs/2502.10954

[4] Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. Do NOT Think That Much for 2+3=? On the Overthinking of o1-Like LLMs. arXiv:2412.21187 [cs.CL] https://arxiv.org/abs/2412.21187

[5] Kaustubh Dhole, Nikhita Vedula, Saar Kuzi, Giuseppe Castellucci, Eugene Agichtein, and Shervin Malmasi. 2025. Generative Product Recommendations for Implicit Superlative Queries. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*. 77–91.

[6] 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* (2024).

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

[8] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Trans. Inf. Syst.* 43, 2, Article 42 (Jan. 2025), 55 pages. doi:10.1145/3703155

[9] Seongmin Lee, Hsiang Hsu, and Chun-Fu Chen. 2024. LLM Hallucination Reasoning with Zero-shot Knowledge Test. arXiv:2411.09689 [cs.AI] https://arxiv.org/abs/2411.09689

[10] Lei Li, Xiao Zhou, and Zheng Liu. 2025. R2MED: A Benchmark for Reasoning-Driven Medical Retrieval. arXiv:2505.14558 [cs.IR] https://arxiv.org/abs/2505.14558

[11] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the Middle: How Language Models Use Long Contexts. arXiv:2307.03172 [cs.CL] https://arxiv.org/abs/2307.03172

[12] Wentao Liu, Hanglei Hu, Jie Zhou, Yuyang Ding, Junsong Li, Jiayi Zeng, Mengliang He, Qin Chen, Bo Jiang, Aimin Zhou, and Liang He. 2025. Mathematical Language Models: A Survey. arXiv:2312.07622 [cs.CL] https://arxiv.org/abs/2312.07622

[13] 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* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 2421–2425. doi:10.1145/3626772.3657951

[14] Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models. arXiv:2410.05229 [cs.LG] https://arxiv.org/abs/2410.05229

[15] Tong Niu, Shafiq Joty, Ye Liu, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. JudgeRank: Leveraging Large Language Models for Reasoning-Intensive Reranking. arXiv:2411.00142 [cs.CL] https://arxiv.org/abs/2411.00142

[16] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375* (2019).

[17] Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. RankZephyr: Effective and Robust Zero-Shot Listwise Reranking is a Breeze! *arXiv preprint arXiv:2312.02724* (2023).

[18] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084 [cs.CL] https://arxiv.org/abs/1908.10084

[19] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 3715–3734.

[20] Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, 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. arXiv:2504.20595 [cs.AI] https://arxiv.org/abs/2504.20595

[21] Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han yu Wang, Haisu Liu, 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. arXiv:2407.12883 [cs.CL] https://arxiv.org/abs/2407.12883

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

[23] Orion Weller, Kathryn Ricci, Eugene Yang, Andrew Yates, Dawn Lawrie, and Benjamin Van Durme. 2025. Rank1: Test-Time Compute for Reranking in Information Retrieval. arXiv:2502.18418 [cs.IR] https://arxiv.org/abs/2502.18418

[24] Eugene Yang, Andrew Yates, Kathryn Ricci, Orion Weller, Vivek Chari, Benjamin Van Durme, and Dawn Lawrie. 2025. Rank-K: Test-Time Reasoning for Listwise Reranking. arXiv:2505.14432 [cs.IR] https://arxiv.org/abs/2505.14432

[25] Shengyao Zhuang, Xueguang Ma, Bevan Koopman, Jimmy Lin, and Guido Zuccon. 2025. Rank-R1: Enhancing Reasoning in LLM-based Document Rerankers via Reinforcement Learning. arXiv:2503.06034 [cs.IR] https://arxiv.org/abs/2503.06034


# 7 Appendix

## 7.1 Effect of shuffling

Tables 7 and 8 illustrate the performance differences while passing in a shuffled order of the documents and their associated scores.

## 7.2 Effect of normalization

Tables 9 and 10 show the effect of normalization on all splits.

## 7.3 Reasoning Traces

We also provide examples of reasoning traces that demonstrate how an LLM (Deepseek-R1) leverages and thinks with BM25 reasoning.

### *7.3.1 Query.*

> How to count reads in bam per bed interval with bedtools/nI recently installed Ubuntu 16.04 (because I was still using 12.04). But it seems my bedtools scripts don't work properly anymore. I can't figure out how to use the new bedtools for my old ways. What I want to do is get the number of reads from a bam file, per interval from a bed file. It was very simple with my old version of Ubuntu and bedtools:bedtools coverage -abam file.bam -b All_peaks.bed > file.cov.txt.But this doesn't seem to work anymore. I used to get small files as results (3MB), but now huge files are created (4GB) including read names?? I am using bedtools v2.25 now.,

### *7.3.2 With BM25 reasoning (on R2MED).*

> ...
>
> Looking at the provided passages, I need to find which ones discuss changes in bedtools coverage behavior...
>
> ...
>
> Looking through the provided documents, Document 2's content about the coverage tool's change in behavior might be relevant.
>
> ...
>
> Looking back at the provided passages, Document 8 mentions the -split option for coverage calculations. Document 21 talks about the -bga option for genomecov to report all positions, which is more verbose. But the user's problem is with coverage, not genomecov.
>
> ...

**Table 7: Effect of shuffling on R2MED**

| Setting | Bioinfo | Biology | IIYi-Clin | Med-Sci | MedQA-Diag | MedXQA | PMC-Clin | PMC-Treat | Avg |
|---|---|---|---|---|---|---|---|---|---|
| Shuffled | **.621** | **.533** | **.150** | **.569** | **.564** | **.369** | **.491** | **.610** | **.488** |
| Shuffled w/ BM25 | .586 | .518 | .084 | .533 | .505 | .313 | .451 | .568 | .445 |

**Table 8: Effect of shuffling on BRIGHT**

| Setting | Biology | Earth Sci | Econ | Psych | Robot | Stack | Sust | LC | Pony | AoPS | TheoQ | TheoT | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Shuffled | .438 | .380 | .254 | .449 | .254 | .220 | .341 | .210 | **.281** | .043 | .220 | .334 | .285 |
| Shuffled w/ BM25 | **.508** | **.444** | **.270** | **.489** | **.296** | **.285** | **.356** | **.255** | .225 | **.077** | **.264** | **.394** | **.322** |

**Table 9: Performance of BM25+Norm with different scaling factors across domains on Gemini benchmark**

| Model | Bio | Earth | Econ | Psy | Robot | Stack | Sust | LC | Pony | AoPS | TheoQ | TheoT | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0-1 scale | .531 | .454 | .302 | .504 | .291 | .363 | .333 | .250 | .314 | .069 | .276 | .394 | .340 |
| 0-100 scale | .541 | .444 | .298 | .518 | .296 | .346 | .356 | .253 | .300 | .077 | .282 | .394 | **.342** |

**Table 10: R2MED normalized BM25 scores**

| Setting | Bioinfo | Biology | IIYi-Clin | Med-Sci | MedQA-Diag | MedXQA | PMC-Clin | PMC-Treat | Avg |
|---|---|---|---|---|---|---|---|---|---|
| 0-1 scale | .588 | .573 | .130 | .560 | .576 | .351 | .465 | .614 | .482 |
| 0-100 scale | **.621** | .533 | **.150** | **.569** | .564 | **.369** | **.491** | .610 | **.488** |

> Looking at Document 1 again: it's about genomecov's options, not coverage.
> ...
> **Let's look at the highest BM25 documents again. Document 2 talks about coverage tool changes, which is relevant**. Document 8 discusses coverage options including -split and -strand. Document 1 is about genomecov.
> ...
> Given the information in the provided passages, the most relevant documents are:
> Document 2: Discusses changes in the coverage tool's behavior in version 2.24.0, including processing order and -sorted option.
> Document 8: Mentions options for coverage calculation, including -split...
> Document 21: Talks about genomecov's -bga option...
> ...
> Given the confusion, the most likely relevant passage is Document 2, which discusses changes in the coverage tool's behavior.
> ...
> **However, considering the BM25 scores, Document 2 has the second highest score and discusses coverage tool changes, which is likely relevant.** Document 8 discusses coverage options and could be related. Document 21 talks about genomecov's -bga option...
>
> In conclusion, the most relevant passages are likely Document 2 and Document 8.

#### 7.3.3 *Without BM25 reasoning on R2MED.*

> The user is using: bedtools coverage -abam file.bam -b All-peaks.bed > file.cov.txt Previously, this produced a 3MB summary file, but now it generates a 4GB file including read names.
> ...
> Document 2 notes a critical change in bedtools version 2.24.0:
> "coverage is now computed for the intervals in the A file based on overlaps with the B file, rather than vice versa."
> That's key.
> In the old version, -abam file.bam -b peaks.bed treated the BED intervals as targets for coverage. In the new version, bedtools computes coverage for A (the BAM) over B (the BED). Since BAM entries are individual reads, the tool now outputs each read along with overlap details — hence the explosion in file size and appearance of read names.
> ...
> The correct fix in v2.25 is:
> bedtools coverage -a All_peaks.bed -b file.bam > file.cov.txt
> ...
> This change in semantics (A vs. B) is the root cause, and is well-documented in Document 2. Other documents (8, 14, 21) mention flags like -split, -d, or -bga but aren't directly relevant here.

#### 7.3.4 *Gold Answer.*

> Uppers" and "downers" are colloquial terms for drugs with stimulant and depressant effects, respectively. ... The primary concern in overdose is overstimulation or suppression of cardiovascular/respiratory systems. ... One combination is a "speedball" (cocaine + heroin/morphine). The mix gives euphoria and masks side effects, so users may take higher doses. ... However, drug durations differ: when the stimulant wears off, opioid effects persist, leading to unexpected respiratory depression and death. ... Alcohol can worsen effects due to paradoxical reactions; alcohol + cocaine is also linked to overdose deaths. ... Stimulant + depressant combos mask warning signs and increase overdose and abuse potential. ...

## 7.4 Full Prompts

### *7.4.1 Full prompts for BRIGHT.*

> The following passages are related to query: cur_query
> {retriever_prompt}{doc_string} First identify the essential problem in the query. Think step by step to reason about why each document is relevant or irrelevant. Rank these passages based on their relevance to the query. Please output the ranking result of passages as a list, where the first element is the id of the most relevant passage, the second element is the id of the second most element, etc. Please strictly follow the format to output a list of {topk} ids corresponding to the most relevant {topk} passages: "'json [...]

### *7.4.2 Full prompts for R2MED.*

> The following passages are related to the query: "cur_query".
> {retriever_prompt}
> {doc_string} First, identify the essential problem or topic in the query. Think step by step to reason about why each document is relevant or irrelevant.. Rank these passages based on their relevance to the query. Please output the ranking result of passages as a list, where the first element is the id of the most relevant passage, the second element is the id of the second most element, etc. Finally, output a ranked list of the top topk most relevant passages by their index number. Please strictly follow the format to output a list of {topk} ids corresponding to the most relevant {topk} passages: "'json [...]

The retriever prompt which introduces BM25 is defined as follows,

> You are also given the BM25 scores from a lexical retriever for each document.
> [1]. {doc_text_1} BM25 score: {score_1}
> [2]. {doc_text_2} BM25 score: {score_2}
> [3]. {doc_text_3} BM25 score: {score_3}
> ...
> First identify the essential problem in the query. Think step by step to reason about why each document is relevant or irrelevant. Rank these passages based on their relevance to the query. Please output the ranking result of passages as a list, where the first element is the id of the most relevant passage, the second element is the id of the second most element, etc. Please strictly follow the format to output a list of topk ids corresponding to the most relevant topk passages: "'json [...]