Lighting the Way for BRIGHT: Reproducible Baselines with Anserini, Pyserini, and RankLLM

Sahel Sharifymoghaddam*
University of Waterloo
David R. Cheriton School of Computer Science
Waterloo, Canada
sahel.sharifymoghaddam@uwaterloo.ca

Yijun Ge*
University of Waterloo
David R. Cheriton School of Computer Science
Waterloo, Canada
l2ge@uwaterloo.ca

Raghav Vasudeva
University of Waterloo
David R. Cheriton School of Computer Science
Waterloo, Canada
r2vasude@uwaterloo.ca

Jimmy Lin
University of Waterloo
David R. Cheriton School of Computer Science
Waterloo, Canada
jimmylin@uwaterloo.ca

Abstract

Retrieval benchmarks for large language models (LLMs) should reflect the long, reasoning-intensive queries typical of retrieval-augmented generation (RAG). We present a systematic study of BRIGHT, a reasoning-focused retrieval benchmark, along with strong, reproducible reference methods integrated into Anserini, Pyserini, and RankLLM. We evaluate lexical, sparse, dense, and fusion-based retrievers, as well as LLM rerankers, under long-query settings. In reproducing BRIGHT's lexical baseline, we identify a key under-documented detail: query-side BM25 (BM25Q), which applies BM25 weighting to the query itself. On long, multi-sentence queries, BM25Q consistently outperforms standard BM25, making it the strongest lexical baseline for reasoning-oriented retrieval. We further audit the BRIGHT corpus, uncovering data quality issues that impact evaluation, and offer mitigation. Finally, we study the generalizability of BM25Q across five additional benchmarks, finding its gains largely specific to BRIGHT, while fusion with standard BM25 provides the most consistent improvements across datasets.

CCS Concepts

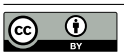
• **Information systems** → **Information retrieval**; **Retrieval models and ranking**.

Keywords

Retrieval, Reranking, BM25, BRIGHT, RRF, NAF, BEIR, MIRACL

ACM Reference Format:
Sahel Sharifymoghaddam, Yijun Ge, Raghav Vasudeva, and Jimmy Lin. 2026. Lighting the Way for BRIGHT: Reproducible Baselines with Anserini, Pyserini, and RankLLM. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26), July 20–24, 2026, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3805712.3808570

*Both authors contributed equally to this research.



This work is licensed under a Creative Commons Attribution 4.0 International License.
SIGIR '26, Melbourne, VIC, Australia
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2599-9/2026/07
https://doi.org/10.1145/3805712.3808570

1 Introduction

The growing use of large language models (LLMs) and retrieval-augmented generation (RAG) [10, 18, 27] is transforming the role of information retrieval. In RAG, an LLM conditions on retrieved documents to generate more accurate and grounded responses, leveraging both proprietary and publicly available data. The BRIGHT benchmark [35] was developed to evaluate retrieval systems specifically for this application, focusing on their ability to handle the complex, reasoning-intensive queries often found in user prompts.

To facilitate further research with the BRIGHT benchmark, we have integrated reproducible baselines into two popular information retrieval (IR) toolkits, Anserini [20, 39] and Pyserini [21], for convenient first-stage retrieval. The baselines we incorporate include BM25 [30], a classic lexical retrieval algorithm, and BGE-large-en-v1.5 [38], a representative dense retrieval model, both from the original BRIGHT work. We also include SPLADE-v3 [17], a representative model for learned sparse retrieval. Finally, we include Diver-Retriever-4B [24] and Reason-Embed-4B [4], two of the top-performing reasoning-aware retrievers on the BRIGHT leaderboard. Additionally, we have incorporated BRIGHT into RankLLM [33], a widely used Python package for second-stage reranking with LLMs, which has been shown to provide significant improvements in retrieval effectiveness [11, 25, 26, 36].

In establishing these baselines, we find that the BM25 results reported in the original BRIGHT paper [35] differ from what we obtained initially using Anserini and Pyserini. Like Lucene [3], by default, Anserini uses bag-of-words (BoW) to generate sparse query vectors, whereas the implementation used by the BRIGHT paper applies the BM25 scoring algorithm to each query token to obtain token weights, a variant we refer to as "query-side BM25" (BM25Q).

Finally, we investigate the generality of the BM25Q technique and assess whether its benefits extend beyond BRIGHT to other standard retrieval benchmarks. To summarize, our five primary contributions are:

- **BM25 vs. BM25Q, Analysis and Insights (New).** We isolate and implement both standard BM25 (BoW queries) and BM25Q, enabling a controlled comparison. We show that BM25Q's improvements are driven by long query effects and characterize when it outperforms standard BM25.

- **Reproducible BRIGHT Baselines (Reproducibility).** We provide fully reproducible implementations of BRIGHT retrieval and reranking pipelines, integrating first-stage baselines into Anserini/Pyserini and enabling listwise reranking via RankLLM.
- **Fusion Strategies on BRIGHT (New).** We present a systematic evaluation of hybrid fusion methods (RRF and NAF) on BRIGHT, demonstrating that simple fusion strategies with BM25Q can surpass the effectiveness of individual retrievers, including widely used neural models.
- **BRIGHT Corpus Audit (New).** We conduct the first systematic audit of the BRIGHT corpora, identifying duplicates and degenerate passages, and quantify their impact on retrieval evaluation.
- **Generalization Beyond BRIGHT (New):** We evaluate BM25, BM25Q, and fusion strategies across five additional benchmark suites, showing that BM25Q gains are not universal and depend strongly on query characteristics.

2 Background and Related Work

BRIGHT. This is a benchmark designed to evaluate how well retrieval systems handle reasoning-intensive tasks, which remains a significant challenge for current state-of-the-art methods [35]. The benchmark consists of multiple heterogeneous corpora spanning several domains, including Wikipedia-style passages, technical and scientific documents, and web-derived content. Overall, BRIGHT contains approximately tens of thousands to hundreds of thousands of documents per corpus and a few thousand queries distributed across tasks, with each query requiring retrieval based on multi-hop reasoning, semantic composition, or document-level understanding rather than simple keyword matching or semantic similarity. Further details on the benchmark construction, task formulation, and dataset statistics can be found in the original paper [35].

Unlike standard IR benchmarks such as BEIR [37] and TREC Deep Learning tracks [5–9], where queries are typically short keyword-like expressions, BRIGHT queries are significantly longer and often resemble natural-language questions or problem statements. Many queries require combining multiple pieces of evidence or reasoning across concepts, making purely lexical matching or shallow semantic similarity insufficient. For example, a query may ask for a theorem that can be applied to solve a given mathematical problem. Answering such a query requires understanding the structure of the problem, identifying relevant mathematical principles, and mapping them to applicable theorems, rather than relying on surface-level lexical overlap or semantic similarity alone.

This difficulty is reflected in BRIGHT's leaderboard,[1] where the highest nDCG@10 scores remain relatively low. A common strategy to improve effectiveness is the use of LLMs, either for query expansion or reranking. This pattern is evident in the leaderboard, where top-performing systems consistently incorporate LLM-based components; notably, systems using LLM-based query expansion achieve nDCG@10 scores more than 10 percentage points higher than those using the original queries [24]. Although BRIGHT offers both original and LLM-expanded queries along with first-stage retrieval runs, the results for the original queries are significantly weaker than those for the LLM-enhanced queries; nevertheless, they remain an essential part of the retrieval pipeline. In this work, we reproduce two widely used first-stage retrieval baselines from the BRIGHT paper using BM25 and BGE-large-en-v1.5 with the original queries.

BM25. This is a family of probabilistic bag-of-words (BoW) ranking functions [30] that originated in Okapi's TREC-3 [14] runs as a compromise between BM11 and BM15 versions to perform lexical matching with sensible term frequency (TF) saturation and length normalization [29]. A widely used BM25 version scores a document D for a query $Q = \{q_i\}$ as:

$$\text{score}(D, Q) = \sum_i \text{IDF}(q_i) \; \frac{(k_1 + 1)\, tf_{i,D}}{tf_{i,D} + k_1\, \ell(D)} \; \frac{(k_3 + 1)\, qtf_i}{k_3 + qtf_i}, \quad (1)$$

where $tf_{i,D}$ is the frequency of q_i in D; $\ell(D) = 1 - b + b\,|D|/\overline{|D|}$ is the length-normalization term; and $IDF(q_i)$ is the inverse document frequency defined as follows (with N and n_i referring to the total number of documents and those containing term q_i):

$$\text{IDF}(q_i) = log \frac{N - n_i + 0.5}{n_i + 0.5} \quad (2)$$

In this formulation, k_1 controls the influence of term frequency saturation, b determines the strength of document length normalization, and k_3 controls how query term frequency (qtf) contributes to the retrieval score. When $k_3 = \infty$, the model reduces to a bag-of-words representation in which term weights increase linearly with frequency, whereas $k_3 = 0$ yields a binary representation that considers only term presence. Early BM25 variants often omitted query term frequency, but later formulations introduced finite values of k_3 to allow repeated query terms to influence scoring [29]. Many modern implementations today adopt the bag-of-words setting, treating qtf linearly and thereby allowing repeated terms to affect retrieval scores [3, 15, 39].

While BM25-style term weighting is well established on the document side, its application on the query side has received comparatively limited attention, with some notable exceptions. Prior work [31] generally assumes that query term frequency has limited impact, reflecting the short length and low repetition typical of keyword queries. Alternative lines of work have explored query reweighting and expansion approaches, such as divergence-from-randomness models [1], which adjust query term importance based on statistics from an initial set of retrieved documents. However, these approaches rely on a first-pass retrieval stage, whereas BM25Q applies term-frequency saturation directly to the query within a single-stage scoring function.

For long queries—such as in document similarity tasks where the query is itself another document [22]—both term-frequency effects and query length normalization become important. This setting is particularly relevant for BRIGHT, where queries are substantially longer than in traditional datasets and frequently contain repeated tokens. To account for these characteristics, the authors of the BRIGHT paper apply BM25-style term-frequency saturation symmetrically to both documents and queries in their baseline formulation, resulting in a "query-as-document" factorization:

$$\text{score}(D, Q) = \sum_{t \in D \cap Q} \text{IDF}(t)^2 \; \frac{(k_1 + 1)\, tf_{t,D}}{tf_{t,D} + k_1\, \ell(D)} \; \frac{(k_1 + 1)\, tf_{t,Q}}{tf_{t,Q} + k_1\, \ell(Q)}, \quad (3)$$

[1] https://brightbenchmark.github.io/

where $\ell(x) = 1 - b + b\,|x|/\overline{|D|}$ uses the *corpus* average length $\overline{|D|}$ for length normalization on both sides. Compared to eq. (1), this formulation places IDF(t) in both vectors, resulting in an IDF(t)^2 term that further emphasizes rare terms while tempering the influence of repeated query tokens. We refer to this variant as *query-side BM25* (BM25Q).

Learned Dense and Sparse Retrieval. These methods have driven the recent advances in information retrieval in two major directions beyond traditional lexical methods. Dense retrieval models [16] encode queries and documents into continuous embedding vectors and compute similarity through inner products. Among these, the BGE [38] family of models has become widely adopted due to its strong performance across diverse benchmarks. In particular, BRIGHT reports results with BGE-large-en-v1.5, a dense model that generates high-quality semantic representations for both queries and documents, enabling retrieval based on meaning rather than exact term matches.

In parallel, learned sparse retrieval models extend the bag-of-words paradigm by introducing neural weighting mechanisms [19]. SPLADE-v3 [17] exemplifies this approach by analyzing query and document tokens and assigning importance weights that capture both lexical and semantic relevance. Importantly, SPLADE-v3 can also expand the representation by introducing semantically related tokens that are not present in the original text. This enables the model to capture richer relationships while retaining the interpretability and efficiency of sparse representations. Together, learned dense and sparse retrievers represent two complementary strategies: dense models capture deep semantic similarity in continuous space, while sparse models refine token-level weighting in discrete space. BRIGHT provides a valuable setting to compare these approaches, since its reasoning-intensive queries stress both semantic understanding and token-level precision.

Listwise Reranking. Advances in instruction-following LLMs have made them effective listwise rerankers [26, 36]. In this setting, reranking is formulated as sorting a list of candidate documents for a given query according to a relevance criterion, and is treated as a generative task in which the LLM produces an ordered list of candidate identifiers. This approach leverages the model's reasoning ability to compare candidates with respect to the query, making it particularly well suited for reasoning-intensive ranking tasks such as BRIGHT.

Reasoning-Focused Retrievers and Rerankers. Recent work on reasoning-intensive IR (e.g., BRIGHT) moves beyond lexical or surface-level semantic matching toward modeling task-relevant helpfulness, where a document may be relevant because it provides the missing concept, rule, or intermediate knowledge needed to solve the query rather than directly paraphrasing the answer. [4, 23, 24, 32] A common recipe is to start from a strong general-purpose embedding or encoder model and post-train it with automatically constructed supervision that encodes reasoning-aware relevance—typically by synthesizing challenging queries from documents and pairing them with plausible-but-unhelpful hard negatives. [4, 32] For reranking, many approaches similarly begin with a generic cross-encoder or LLM reranker and train it on reasoning-intensive, domain-diverse data (often labeled or filtered by a strong reasoning model), sometimes combining supervised fine-tuning with reinforcement learning or other post-training techniques to better capture multi-step relevance signals [4, 23, 24].

Table 1: Retrieval effectiveness (nDCG@10) of BM25 variants on BRIGHT ($k_1 = 0.9, b = 0.4$). We compare Bag-of-Words (BoW) and query-side BM25 (BM25Q) using default bucketized length normalization against their accurate (A.) counterparts. Original BRIGHT (Orig.) numbers are included as a baseline. All values are percentages.

Dataset	Anserini				Orig.
	(a) BoW	(b) BoW A.	(c) BM25Q	(d) BM25Q A.	(e) BM25Q
Stack Exchange					
Biology	18.2	17.5	19.7	18.9	18.9
Earth Science	27.9	27.0	27.9	27.2	27.2
Economics	16.5	15.9	15.2	14.8	14.9
Psychology	13.4	12.9	12.7	12.6	12.5
Robotics	10.9	10.7	13.9	13.7	13.6
Stack Overflow	16.3	16.6	18.6	18.5	18.4
Sust. Living	16.1	16.2	15.2	15.0	15.0
Average	**17.0**	**16.7**	**17.6**	**17.2**	**17.2**
Coding					
LeetCode	24.7	24.6	25.0	24.4	24.4
Pony	4.3	4.0	7.9	7.7	7.9
Average	**14.5**	**14.3**	**16.5**	**16.1**	**16.2**
Theorem					
AoPS	6.5	6.4	6.3	6.2	6.2
TheoremQA - Q	7.3	7.5	10.4	10.4	10.4
TheoremQA - T	2.1	2.1	4.9	4.9	4.9
Average	**5.3**	**5.3**	**7.2**	**7.2**	**7.2**
Overall Avg.	**13.7**	**13.5**	**14.8**	**14.5**	**14.5**

3 Experimental Setup

Our experiments employ Anserini [20, 39] and Pyserini [21] for first-stage retrieval and RankLLM [33] for second-stage reranking, enabling fully reproducible experimental pipelines. We select these toolkits—and provide direct integrations with them—because they are widely adopted in the IR community and support streamlined, two-click reproducibility.[2] Documentation and usage instructions are available in the respective GitHub repositories,[3,4,5] and prebuilt indexes are released on HuggingFace.[6] For all experiments, we utilize Pyserini's default BM25 parameters (k_1=0.9, b=0.4). We abstain from tuning these parameters as the BRIGHT dataset does not provide a dedicated development split for parameter selection. We evaluate effectiveness using nDCG@10, which is the metric used for the BRIGHT leaderboard. The remainder of this section describes the setup for each of our experiments.

[2] https://castorini.github.io/pyserini/2cr/bright.html
[3] https://github.com/castorini/anserini/tree/master/docs/reproduce/from-document-collection
[4] https://github.com/castorini/pyserini/tree/master/scripts/bright
[5] https://github.com/castorini/rank_llm/tree/main/src/rank_llm/demo
[6] https://huggingface.co/datasets/castorini/prebuilt-indexes-bright

Table 2: First-stage retrieval effectiveness (nDCG@10; reported as ×100) on BRIGHT with BM25Q as the lexical baseline. Neural retrievers include generic SPLADE-v3 (S-v3) and BGE-large-en-v1.5 (BGE), as well as reasoning-focused retrievers Diver-Retriever-4B (Diver) and Reason-Embed-4B (R-Em.). Results are categorized into (1) individual retrievers, (2) RRF fusion with BM25Q, and (3) NAF fusion with BM25Q.

Dataset	Base	(1) Individual Retrievers				(2) RRF with BM25Q				(3) NAF with BM25Q			
	BM25Q	S-v3	BGE	Diver	R-Em.	S-v3	BGE	Diver	R-Em.	S-v3	BGE	Diver	R-Em.
Stack Exchange													
Biology	19.7	21.0	12.4	42.1	54.5	22.0	17.5	35.7	52.8	23.7	20.3	34.6	38.2
Earth Science	27.9	26.7	25.5	46.8	54.0	29.9	31.2	46.1	53.0	30.9	31.6	45.1	48.8
Economics	15.2	16.0	16.6	22.4	34.4	17.0	18.0	27.0	31.2	16.5	19.2	21.7	26.9
Psychology	12.7	15.3	18.1	34.4	46.1	15.8	18.5	32.9	43.8	16.5	18.5	28.7	34.8
Robotics	13.9	15.8	12.3	21.5	34.6	17.0	15.4	26.2	28.0	15.7	15.3	21.2	27.4
Stack Overflow	18.6	12.9	11.0	20.9	35.9	17.5	18.2	30.1	29.4	18.6	20.5	25.8	33.4
Sust. Living	15.2	15.0	14.4	25.1	37.0	15.2	15.8	26.5	31.0	16.4	16.5	23.2	28.6
Average	**17.6**	**17.5**	**15.7**	**30.4**	**42.4**	**19.2**	**19.2**	**32.1**	**38.4**	**19.8**	**20.3**	**28.6**	**34.0**
Coding													
LeetCode	25.0	26.0	26.7	37.8	37.1	28.2	29.3	40.6	38.0	28.8	30.1	39.3	41.9
Pony	7.9	14.4	3.4	12.9	12.0	16.3	15.7	14.8	25.6	15.5	10.6	15.1	14.7
Average	**16.5**	**20.2**	**15.0**	**25.4**	**24.6**	**22.3**	**22.5**	**27.7**	**31.8**	**22.1**	**20.4**	**27.2**	**28.3**
Theorem													
AoPS	6.3	6.9	6.4	10.3	11.3	8.3	7.0	12.9	12.5	8.6	7.6	12.3	13.1
TheoremQA - Q	10.4	11.1	14.1	37.7	40.7	11.1	12.7	25.5	39.9	11.9	14.5	30.1	32.1
TheoremQA - T	4.9	5.5	5.3	38.0	45.5	7.7	8.8	24.2	45.5	7.3	8.1	29.0	32.9
Average	**7.2**	**7.9**	**8.6**	**28.6**	**32.5**	**9.0**	**9.5**	**20.9**	**32.6**	**9.3**	**10.1**	**23.8**	**26.1**
Overall Avg.	**14.8**	**15.6**	**13.8**	**29.1**	**36.9**	**17.2**	**17.3**	**28.5**	**35.9**	**17.5**	**17.7**	**27.2**	**31.1**

3.1 Isolating BM25 Implementation Choices

To quantify the impact of lexical retrieval configurations on long, reasoning-heavy queries, we examine two orthogonal implementation dimensions:

- **Query Representation.** We compare the standard *bag-of-words* (BoW) implementation—where query term frequencies are treated linearly—against *query-side BM25* (BM25Q). The latter applies BM25-style term-frequency saturation and length normalization to the query itself (treating it as a "document" per Equation 3).
- **Document-Length Normalization Precision.** Lucene's default `BM25Similarity` optimizes scoring speed by quantizing document-length norms into a single byte, using a 256-value lookup table for normalization factors [3, 39]. While this approximation typically yields negligible differences in effectiveness [15], we compare it against Anserini's exact-length implementation to assess its impact on reproducibility. To ensure a faithful comparison with the BRIGHT baseline, we include the exact-length version for reference but maintain the Lucene default for consistency with broader Anserini benchmarks.

Both of these configurations can be toggled via flags in the indexing and retrieval commands.

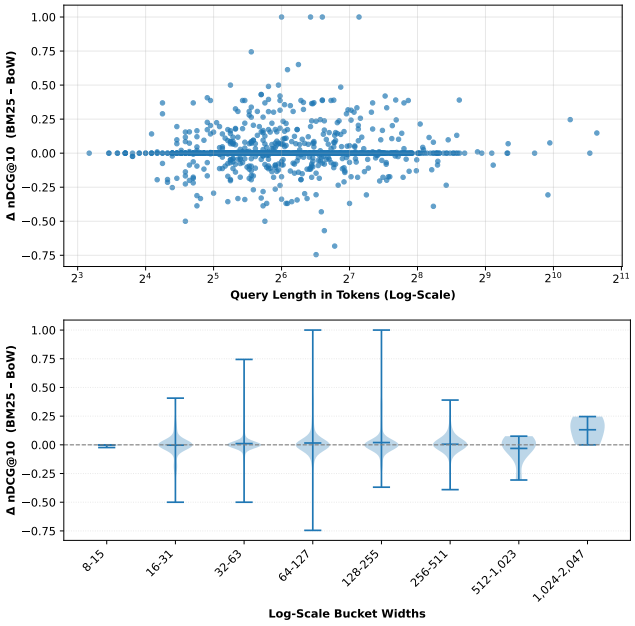


Figure 1: Query length (in tokens) vs. ΔnDCG@10 (BM25Q - BoW) for all queries in BRIGHT tasks.

3.2 BRIGHT Reference Methods

We establish reproducible reference methods for BRIGHT spanning three paradigms to evaluate lexical, semantic, and reasoning-centric capabilities in different stages of the retrieval pipeline:

- **First-Stage Retrieval.** We evaluate representative retrievers from major families: BM25 (lexical), BGE-large-en-v1.5 (dense), and SPLADE-v3 (learned sparse). Furthermore, we include Diver-4B and Reason-Embed-4B, which utilize reasoning-enhanced contrastive fine-tuning and currently rank among the top-3 on the BRIGHT leaderboard as of February 12, 2026.
- **Second-Stage Reranking.** We rerank the top-100 candidates using listwise rerankers via the RankLLM framework. This includes out-of-the-box open-source models—Qwen3-8B and gpt-oss-120b—as well as ReasonRank-32B, a model specifically optimized for reasoning-intensive listwise reranking through reinforcement learning from ranking-based rewards. To accommodate BRIGHT's long-form queries and document snippets, we set the reranking context window to 16k tokens. We use the default window size of 20 and stride of 10 for the sliding window algorithm [36]. For ReasonRank, we use their original ranking prompt[7] as is [23]. The prompt used for the out-of-the-box models is shown in Figure 2. Here, inspired by the ReasonEmbed's custom query instructions per BRIGHT task, we have slightly altered the default RankZephyr [26] prompt template to include task-specific relevance criteria.
- **Fusion.** To leverage complementary retrieval signals, we consider both *Reciprocal Rank Fusion* (RRF) and *Normalized Average Fusion* (NAF) of individual retriever results with BM25Q. For the second-stage reranking input, we adopt a "best-of-first-stage" strategy: we compare the effectiveness of individual retrievers against their fused combinations and select the result with the best nDCG@10 (solo or fused) to serve as the candidate set for RankLLM. This ensures that for each retriever, the reranker is evaluated on the most promising pool of candidates available from the first-stage.

3.3 Generalizability Study Beyond BRIGHT

To test whether BM25Q's behavior is specific to BRIGHT or persists across standard IR workloads, we conduct a head-to-head comparison of BoW, BM25Q, and their tuning-light fusions—RRF and NAF—across five additional benchmark suites: BEIR [37], CURE v1 [34], TREC Disks 1 & 2 [12–14], MIRACL [40], and TREC DL19–DL23 tracks [5–9] with MS MARCO passage retrieval (v1 and v2) [2]. For this experiment, we look at the statistical significance of the results via bootstrap sampling. Exact permutations are used for per-suite tests, while Monte Carlo sampling is used for the all-tasks pooled analysis. Using this technique, we compute 95% *confidence intervals* (CIs) and treat $p > 0.005$ as *not statistically significant* (NS).

4 Experimental Results

4.1 BM25 Implementation Analysis

Table 1 shows nDCG@10 for different variations of BM25 implementations. Notably, we are able to replicate the BM25 results from the original BRIGHT paper almost exactly using the Anserini toolkit; see columns (d) and (e).

Quantized vs. Accurate Length Normalization. Looking at columns (a) vs. (b) and (c) vs. (d) in Table 1, we find that quantized and exact document-length normalization yield nearly identical results. On average, the quantized variant, which is the Lucene default, scores slightly higher by approximately 0.2–0.3 nDCG@10 *percentage points* (PP). Analysis of the document statistics in Table 3 reveals two general patterns:

- **Discrimination Loss:** For tasks where document lengths are tightly clustered, such as Stack Overflow and TheoremQA - Q, accurate normalization performs slightly better. This is because Lucene's single-byte quantization collapses nearby lengths into the same bucket, reducing the model's ability to discriminate between documents.
- **Penalty Mitigation:** In collections with a heavy long tail of documents, such as Biology or Earth Science, quantization often helps by softening the BM25 length penalty. This mitigates the tendency to over-penalize extremely long documents, acting as a form of implicit regularization.

These findings support existing conclusions that quantization has a negligible impact on overall effectiveness. We utilize the quantized variant moving forward for consistency with Anserini defaults.

Table 3: Document and query length statistics (token counts measured with the Pyserini analyzer [21]) for BRIGHT.

Dataset	Document Length				Query Length			
	min	max	mean	stdev	min	max	mean	stdev
Stack Exchange								
Biology	0	3,325	37.2	46.4	14	301	63.1	44.4
Earth Science	0	25,416	40.6	163.8	9	179	55.2	34.2
Economics	0	4,138	44.0	106.7	21	241	87.9	50.4
Psychology	0	26,064	41.8	190.6	15	237	78.5	44.1
Robotics	0	3,692	31.7	73.0	19	1,594	199.8	282.3
Stack Overflow	0	843	117.8	143.2	21	845	135.7	103.5
Sust. Living	0	18,756	38.8	143.2	15	309	81.1	54.3
Coding								
LeetCode	7	12,231	85.4	110.0	53	493	176.0	77.2
Pony	0	246	33.5	24.0	21	117	45.6	18.2
Theorem								
AoPS	8	1,125	110.1	82.5	12	195	43.6	23.9
TheoremQA - Q	8	1,125	110.1	82.5	11	141	51.5	19.3
TheoremQA - T	7	2,706	118.4	128.2	23	91	50.9	16.7

```
system_message: "You are RankLLM, an intelligent assistant that ranks passages based on a
defined criterion."

prefix:
  """I will provide you with {num} passages, each identified by an alphabetical label [].
  Given a Sustainable Living post as a query, rank the passages by how helpful they are for
  answering the query: {query}."""

body:
  """- Passage [{rank}]: {candidate}"""

suffix:
  """Search Query: {query}
  Rank the {num} passages above by their usefulness in answering the query. Include all
  passages and list them using their identifiers in descending order of usefulness.
  Output format: [] > [] (e.g., [B] > [A]).
  Respond only with the ranking and do not include any explanation or extra words."""

output_validation_regex: r"^\[[A-Z]\]( > \[[A-Z]\])*$"

output_extraction_regex: r"\[([A-Z])\]"
```

Figure 2: Inference prompt for listwise reranking with out-of-the-box LLMs.

[7] https://github.com/8421BCD/ReasonRank

Table 4: Second-stage reranking effectiveness (nDCG@10, ×100) on BRIGHT with SPLADE-v3 (S-v3) and BGE-large-en-v1.5 (BGE) as generic retrievers with BM25Q NAF fusions, and reasoning-focused Diver-Retriever-4B (Diver) and Reason-Embed-4B (R-Em). Top-100 candidates are reranked using Qwen3-8B (Qw.), gpt-oss-120b (GPT), and ReasonRank-32B (R-Ra) rerankers.

Dataset	S-v3 (w. BM25Q NAF)				BGE (w. BM25Q NAF)				Diver				R-Em.			
	Base	Qw.	GPT	R-Ra.	Base	Qw.	GPT	R-Ra.	Base	Qw.	GPT	R-Ra.	Base	Qw.	GPT	R-Ra.
Stack Exchange																
Biology	23.7	41.6	44.0	42.4	20.3	36.6	43.1	40.1	42.1	53.8	54.9	56.1	54.5	55.6	55.7	58.4
Earth Science	30.9	45.4	48.9	47.4	31.6	48.2	46.5	48.2	46.8	50.5	53.5	51.6	54.0	50.0	54.5	51.3
Economics	16.5	27.8	32.9	31.2	19.2	28.0	33.0	34.5	22.4	31.4	38.5	35.3	34.4	34.5	39.1	40.0
Psychology	16.5	38.7	40.6	41.3	18.5	39.2	36.6	39.2	34.4	48.6	51.6	53.9	46.1	47.9	52.6	51.8
Robotics	15.7	32.3	36.9	36.4	15.3	25.2	33.7	30.5	21.5	27.1	29.9	34.9	34.6	32.8	37.0	39.2
Stack Overflow	18.6	29.0	31.7	30.7	20.5	30.2	32.0	34.1	20.9	28.1	34.2	33.4	35.9	32.7	38.6	37.4
Sust. Living	16.4	34.7	40.7	37.8	16.5	33.0	37.9	39.1	25.1	40.0	45.2	42.7	37.0	44.3	49.0	45.6
Average	**19.8**	**35.6**	**39.4**	**38.2**	**20.3**	**34.3**	**37.5**	**38.0**	**30.4**	**39.9**	**44.0**	**44.0**	**42.4**	**42.5**	**46.6**	**46.2**
Coding																
LeetCode	28.8	27.1	32.7	25.3	30.1	28.1	33.8	28.3	37.8	20.6	26.0	19.8	37.1	19.4	25.3	21.1
Pony	15.5	20.1	35.7	23.5	10.6	14.6	29.3	16.2	12.9	22.8	32.6	20.8	12.0	12.6	28.5	13.9
Average	**22.1**	**23.6**	**34.2**	**24.4**	**20.4**	**21.4**	**31.6**	**22.3**	**25.4**	**21.7**	**29.3**	**20.3**	**24.6**	**16.0**	**26.9**	**17.5**
Theorem																
AoPS	8.6	11.2	11.9	12.0	7.6	8.7	11.6	8.7	10.3	10.2	11.6	10.2	11.3	9.4	12.5	9.4
TheoremQA - Q	11.9	22.9	24.1	24.3	14.5	23.4	25.7	24.5	37.7	39.8	41.4	41.0	40.7	40.3	42.0	42.2
TheoremQA - T	7.3	20.6	20.4	18.3	8.1	19.8	19.9	20.6	38.0	44.6	49.4	46.3	45.5	49.3	48.4	45.8
Average	**9.3**	**18.2**	**18.8**	**18.2**	**10.1**	**17.3**	**19.1**	**17.9**	**28.6**	**31.5**	**34.1**	**32.5**	**32.5**	**33.0**	**34.3**	**32.5**
Overall Avg.	**17.5**	**29.3**	**33.4**	**30.9**	**17.7**	**27.9**	**31.9**	**30.3**	**29.1**	**34.8**	**39.1**	**37.2**	**36.9**	**35.7**	**40.3**	**38.0**

Bag-of-Words vs. BM25Q Query Representation. When comparing columns (a) and (c) in Table 1, we find that BM25Q query vectors are more effective than plain BoW vectors on *seven* of the twelve datasets, BoW is better on *four*, and the two approaches tie on one. From the aggregate query-length statistics (Table 3) we do not observe a direct correlation between a task's query-length distribution and whether BM25Q or BoW is superior. However, a per-query analysis (Figure 1) reveals three consistent patterns:

- For *short* queries (< 16 tokens) the two methods are almost identical, with BoW only rarely ahead. This is consistent with the traditional practice of *not* applying BM25 to queries [28, 29].
- As the query length grows, both the *frequency* and the *magnitude* of the differences increase up to a peak, then taper off; the decline occurs because the length-normalization term in BM25 begins to dominate for very long queries.
- Either method can win depending on whether down-weighting frequent terms in favor of rarer ones helps or harms the query's key terms. As BM25Q's wins are both *larger* and *more numerous* at medium lengths (Figure 1), we recommend applying BM25Q weighting to queries of roughly 16–256 tokens.

4.2 BRIGHT Reference Results

Retrieval. Table 2 summarizes first-stage retrieval effectiveness (nDCG@10; reported as ×100) on BRIGHT using query-side BM25 as the lexical baseline, with SPLADE-v3 (S-v3) and BGE-large-en-v1.5 (BGE) as generic neural retrievers and Diver-Retriever-4B (Diver) and Reason-Embed-4B (R-Em.) as reasoning-focused retrievers. Results are organized into (1) individual retrievers, (2) RRF fusion with BM25Q, and (3) NAF fusion with BM25Q. To ensure the technical integrity of our experimental setup, we compare the BM25 and BGE results from the original BRIGHT paper, as well as the Diver and R-Em. results from their respective original works. While all other results in Table 2 are new, these four reproduced baselines demonstrate strong agreement with published figures, showing a maximum absolute difference of only 0.3 PP. This close alignment validates the correctness of our implementation and provides a reliable foundation for our subsequent fusion analysis.

In the *individual* setting (group 1), generic retrievers exhibit category-dependent strengths rather than a uniform winner. On Stack Exchange, BGE underperforms both BM25 and S-v3 on average (15.7 vs. 17.6/17.5), while S-v3 is competitive with BM25 overall (15.6 vs. 14.8 overall average) and clearly strongest among the generic models in Coding (20.2 vs. 16.5 BM25 and 15.0 BGE). As expected, the reasoning retrievers are substantially stronger than all generic baselines across categories: Diver and R-Em. achieve overall averages of 29.1 and 36.9, respectively, with R-Em. leading Diver in every subset except Coding, where they are nearly tied on average (25.4 vs. 24.6).

Fusion with BM25Q reveals a clear asymmetry between generic and reasoning retrievers. For S-v3 and BGE, *both* fusion schemes consistently improve over the corresponding individual retriever, indicating complementary lexical–neural signals when the components are of comparable strength. In contrast, fusing BM25Q with reasoning retrievers generally *reduces* effectiveness relative to Diver and R-Em. alone, with the notable exception of Coding

Table 5: Per-task counts of unique, short (<5 tokens), and zero-token document chunks in the BRIGHT dataset.

Dataset		Unique		Short		Zero Length	
	Total	Count	(%)	Count	(%)	Count	(%)
Stack Exchange							
Biology	57,359	49,434	86.2%	7,534	13.1%	329	0.6%
Earth Science	121,249	117,633	97.0%	5,182	4.3%	44	0.0%
Economics	50,220	40,594	80.8%	13,357	26.6%	748	1.5%
Psychology	52,835	43,756	82.8%	13,802	26.1%	780	1.5%
Robotics	61,961	40,431	65.3%	22,617	36.5%	2,229	3.6%
Stack Overflow	107,081	66,270	61.9%	15,749	14.7%	1,006	0.9%
Sust. Living	60,792	50,142	82.5%	15,777	26.0%	613	1.0%
Coding							
LeetCode	413,932	413,932	100.0%	0	0.0%	0	0.0%
Pony	7,894	6,183	78.3%	98	1.2%	2	0.0%
Theorem							
AoPS	188,002	176,508	93.9%	0	0.0%	0	0.0%
TheoremQA - Q	188,002	176,508	93.9%	0	0.0%	0	0.0%
TheoremQA - T	23,839	23,839	100.0%	0	0.0%	0	0.0%

(where lexical cues such as API functions/library names and exact identifiers can rescue misses).

Comparing fusion operators (categories 2 vs. 3), NAF tends to be slightly better for the generic retrievers, whereas RRF is markedly better for the reasoning retrievers. A plausible explanation is that when combining similarly capable systems (e.g., BM25 and generic neural retrievers), score normalization and averaging can preserve complementary high-confidence signals from each method in the final top-10 results. However, when one component is substantially weaker or differently calibrated (e.g., BM25 vs. reasoning-focused retrievers), score-based fusion can overpromote lexically plausible but semantically irrelevant candidates. In contrast, rank-based RRF mitigates this effect by limiting the influence of any single list: documents absent from the second list are treated as ranked last + 1, reducing the likelihood that spurious BM25 "confidence" displaces genuinely strong reasoning-based hits at the top of the ranking.

Reranking. Table 4 reports second-stage reranking effectiveness (nDCG@10×100) when reranking the top-100 candidates using three listwise rerankers: Qwen3-8B (Qw.), gpt-oss-120b (GPT), and ReasonRank-32B (R-Ra.), applied on the same retrievers used in the first stage. For S-v3 and BGE, reranking is applied on their BM25Q NAF fusion results.

Reranking substantially improves effectiveness, particularly for weaker first-stage retrievers. GPT improves S-v3 from 17.5 to 33.4 (+15.9 PP) and BGE from 17.7 to 31.9 (+14.2 PP). In contrast, gains are smaller for stronger reasoning-focused retrievers: Diver improves from 29.1 to 39.1 (+10.0 PP), and Reason-Embed-4B (R-Em.) improves from 36.9 to 40.3 (+3.4 PP). This demonstrates diminishing returns from reranking as first-stage retrieval quality improves.

Comparing rerankers, GPT consistently achieves the strongest overall effectiveness, yielding the highest average nDCG@10 for all retrievers. In comparison with GPT, R-Ra. is particularly effective on Stack Exchange tasks, outperforming GPT in domains such as Biology and Robotics. Given their comparable sizes (32B dense vs. 5.1B active parameters out of 120B MoE) this improvement suggests that R-Ra. is benefiting from its reasoning-aware training. Qw. is consistently the weakest reranker, though it still improves weaker retrievers (e.g., +11.8 PP on S-v3).

Table 6: nDCG@10 under original and adjusted qrels (with added missing gold documents) for Stack Exchange tasks. First-stage retrieval uses SPLADE-v3, BGE-large-en-v1.5, Diver-Retriever-4B, and Reason-Embed-4B as retrievers; second-stage reranking uses gpt-oss-120b.

Task	S-v3		BGE		Diver		R-Em.	
	Orig.	Adj.	Orig.	Adj.	Orig.	Adj.	Orig.	Adj.
Retrieval								
Biology	23.7	23.8	20.3	20.4	42.2	42.9	54.5	55.6
Earth Science	30.9	31.1	31.6	31.8	46.2	46.5	54.1	54.5
Economics	16.5	16.5	19.2	19.2	21.9	21.9	34.4	34.4
Psychology	16.5	16.5	18.5	18.5	34.2	34.2	46.1	46.1
Robotics	15.7	15.7	15.3	15.3	21.3	21.3	34.6	34.6
Stack Overflow	18.6	19.1	20.5	21.0	20.7	21.4	36.2	37.6
Sust. Living	16.4	16.4	16.5	16.4	24.8	24.8	36.9	36.8
Average	**19.8**	**19.9**	**20.3**	**20.4**	**30.2**	**30.4**	**42.4**	**42.8**
Reranking with gpt-oss-120b								
Biology	44.0	44.8	43.1	43.8	54.9	56.2	55.7	56.7
Earth Science	48.9	49.1	46.5	46.7	53.5	54.1	54.5	55.1
Economics	32.9	32.9	33.0	33.0	38.5	38.5	39.1	39.1
Psychology	40.6	40.6	36.6	36.6	51.6	51.6	52.6	52.6
Robotics	36.9	36.9	33.7	33.7	29.9	29.9	37.0	37.1
Stack Overflow	31.7	32.5	32.0	33.4	34.2	35.0	38.6	39.9
Sust. Living	40.7	40.6	37.9	37.8	45.2	45.1	49.0	49.0
Average	**39.4**	**39.6**	**37.5**	**37.8**	**44.0**	**44.3**	**46.6**	**47.1**

Importantly, reranking can degrade performance when applied to strong reasoning-focused retrievers using weaker rerankers. Applying Qw. to R-Em. reduces the average effectiveness by −1.2 PP, indicating that weaker rerankers may disrupt high-quality initial rankings. Sharper degradations are observed in specific domains, such as LeetCode (−17.7 PP). At the domain level, the largest gains occur on reasoning-intensive tasks. For example, GPT improves Diver on TheoremQA-T by +11.4 PP and on Stack Exchange Psychology by +17.2 PP. In contrast, improvements are smaller on coding tasks, where strong retrievers already achieve competitive effectiveness. Especially for LeetCode, regardless of the LLM choice, reranking degrades the results of strong reasoning-focused retrievers. Overall, these results show that listwise LLM reranking provides large gains for weaker retrievers but smaller gains for strong reasoning-focused retrievers, and can degrade performance when reranker capability is insufficient relative to first-stage retrieval quality. This is more common when the retriever is fine-tuned on domain-specific data such as math and coding.

4.3 BRIGHT Corpora

During our experiments, we identified three data quality issues in the BRIGHT corpora.

- **Duplicate Documents.** As shown in Table 5, a substantial proportion of documents are exact duplicates after trimming leading and trailing whitespace, accounting for up to 38% in some tasks.
- **Very Short Documents.** Eight of the twelve tasks contain documents with fewer than five tokens (ranging from 1.2% to 36.5%),

Table 7: Average nDCG@10 and Recall@100 for BoW, BM25Q, and their fusions (RRF and NAF) across six benchmarks, together with statistical comparisons against BoW. All effectiveness values and mean differences are reported on a percentage scale (without the % symbol). We report mean differences ($\overline{\Delta}$), 95% confidence intervals (CI), and p-values for both metrics. The *All tasks* row corresponds to a pooled analysis over all 75 tasks, rather than an average of the six benchmark-level averages.

Dataset	N	Baseline	New	nDCG@10		$\overline{\Delta(N-B)}$ [95% CI]	p	Recall@100		$\overline{\Delta(N-B)}$ [95% CI]	p
				Base	New			Base	New		
BEIR	18	BoW	BM25Q	42.5	40.9	−1.6 [−2.5, −0.7]	0.002	58.7	58.1	−0.6 [−1.2, 0.1]	0.082
		BoW	RRF	42.5	42.1	−0.4 [−0.8, 0.1]	0.134	58.7	59.1	0.4 [0.1, 0.8]	0.040
		BoW	NAF	42.5	42.5	0.0 [−0.5, 0.4]	0.863	58.7	59.1	0.4 [0.0, 0.9]	0.067
BRIGHT	12	BoW	BM25Q	13.7	14.8	1.1 [0.1, 2.1]	0.056	34.7	39.0	4.2 [2.2, 6.3]	0.003
		BoW	RRF	13.7	14.7	1.0 [0.6, 1.4]	0.002	34.7	38.4	3.7 [2.6, 4.7]	0.001
		BoW	NAF	13.7	14.8	1.1 [0.7, 1.5]	0.002	34.7	38.1	3.4 [2.3, 4.4]	0.001
CURE v1	10	BoW	BM25Q	34.3	33.4	−0.9 [−1.4, −0.4]	0.017	47.7	48.7	1.0 [0.4, 1.7]	0.022
		BoW	RRF	34.3	34.4	0.1 [−0.2, 0.3]	0.520	47.7	48.7	1.0 [0.7, 1.4]	0.003
		BoW	NAF	34.3	34.6	0.3 [0.1, 0.5]	0.050	47.7	48.9	1.2 [0.9, 1.6]	0.003
Disks 1 and 2	9	BoW	BM25Q	48.1	48.8	0.7 [−1.4, 2.7]	0.536	18.1	18.9	0.8 [−0.6, 2.1]	0.314
		BoW	RRF	48.1	49.9	1.8 [0.8, 2.9]	0.006	18.1	19.0	0.8 [0.1, 1.6]	0.080
		BoW	NAF	48.1	49.7	1.6 [0.7, 2.5]	0.014	18.1	19.1	0.9 [0.1, 1.7]	0.072
MIRACL	18	BoW	BM25Q	38.5	36.3	−2.2 [−3.4, −0.6]	0.010	77.2	75.3	−2.0 [−3.2, −0.6]	0.012
		BoW	RRF	38.5	37.8	−0.7 [−1.4, 0.2]	0.100	77.2	77.6	0.4 [−0.2, 1.0]	0.313
		BoW	NAF	38.5	38.2	−0.3 [−1.0, 0.7]	0.594	77.2	77.5	0.3 [−0.3, 1.0]	0.489
MS MARCO	8	BoW	BM25Q	29.6	28.7	−0.9 [−1.5, −0.3]	0.043	36.9	37.0	0.1 [−0.9, 1.7]	0.961
		BoW	RRF	29.6	29.7	0.1 [−0.2, 0.4]	0.681	36.9	37.7	0.8 [0.3, 1.4]	0.019
		BoW	NAF	29.6	29.7	0.1 [−0.3, 0.7]	0.650	36.9	37.8	0.9 [0.2, 1.7]	0.027
All tasks	**75**	**BoW**	**BM25Q**	**35.1**	**34.3**	**−0.8 [−1.4, −0.2]**	**0.005**	**50.6**	**51.0**	**0.4 [−0.4, 1.0]**	**0.408**
		BoW	**RRF**	**35.1**	**35.3**	**0.2 [−0.2, 0.5]**	**0.441**	**50.6**	**51.7**	**1.1 [0.7, 1.5]**	**< 0.001**
		BoW	**NAF**	**35.1**	**35.5**	**0.4 [0.0, 0.7]**	**0.026**	**50.6**	**51.7**	**1.1 [0.7, 1.5]**	**< 0.001**

and some documents are even empty. These cases typically arise when documents consist primarily of whitespace that is removed during tokenization.

We therefore recommend additional cleaning of scraped documents prior to chunking them into shorter passages. This step is particularly important for reasoning-intensive retrievers fine-tuned on math and coding tasks, which are sensitive to formatting and spacing artifacts. For example, removing trailing whitespace from document `aqua_rat_75207` in the AoPS corpus in Pyserini reduces the cosine similarity between embeddings produced by Pyserini and the original Diver codebase[8] from 0.9998 to 0.8700, illustrating the substantial impact that minor formatting differences can have on embedding representations.

- **Missing Query Relevance.** Most duplicate documents correspond to low-quality fragments that are unrelated to any query and thus have limited impact on retrieval evaluation. However, in some cases, relevant gold document IDs are absent from the positive candidate lists; restoring these missing duplicates can change the measured results. To quantify the effect of missing duplicate IDs, we modified the relevance labels (qrels) as follows: for each set of identical documents, if any document ID was marked relevant for a given query, we marked all document IDs in that set as relevant.

[8]https://github.com/AQ-MedAI/Diver

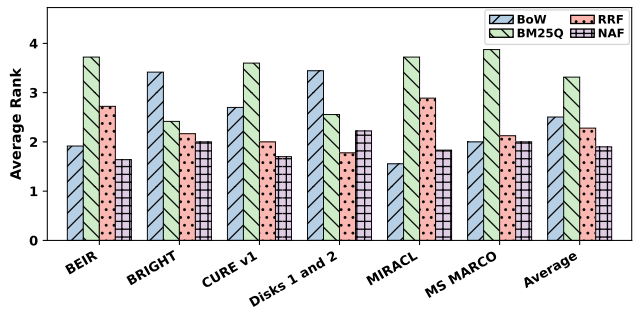
Table 6 compares nDCG@10 scores for retrieval and reranking using the original BRIGHT qrels versus the adjusted qrels for the Stack Exchange category. The Coding and Theorem categories are excluded from this analysis, as they contain no missing gold IDs.

For first-stage retrieval, the average impact is modest, increasing nDCG@10 by 0.1–0.4 PP for different retrievers. As expected, stronger retrievers are more heavily impacted by the original qrels; because they are more effective at surfacing relevant documents, they are disproportionately penalized when those documents are incorrectly treated as negatives. This trend is mirrored in the maximum per-task differences: 0.5 PP for BGE and S-v3, compared to 0.7 for Diver and 1.4 for R-Em.

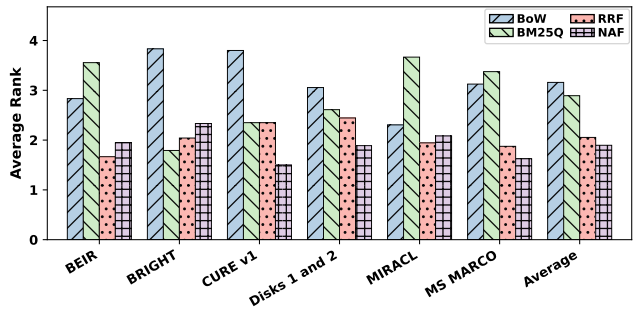
In second-stage reranking, the average nDCG@10 discrepancies increase slightly (0.2–0.5 PP) as the reranker further promotes truly relevant documents to the top of the list. To mitigate this evaluation bias, we recommend removing duplicate documents from the dataset. In the meantime, we have integrated these adjusted qrels into Pyserini to support more accurate benchmarking.

5 BM25Q Generalizability

Table 7 reports the average nDCG@10 and Recall@100 for all tasks across six benchmarks: BEIR, BRIGHT, CURE v1, TREC Disks 1 & 2, MIRACL, and TREC DL19-D23 tracks and their dev sets. BoW and BM25Q yield 35.1 vs. 34.3 average nDCG@10, while the fusions



(a) nDCG@10



(b) Recall@100

Figure 3: Average ranking of BoW, BM25Q, and their fusions (RRF and NAF) across six benchmarks, measured by (a) nDCG@10 and (b) Recall@100.

are slightly higher (RRF 35.3, NAF 35.5). For Recall@100, BM25Q is modestly above BoW (51.0 vs. 50.6), and both fusions attain 51.7. Per-suite patterns align with these averages: BM25Q lags behind BoW on nDCG in BEIR, CURE v1, MIRACL, and TREC DL tracks, but is ahead in the remaining two. For recall, the trend is reversed with BM25Q being ahead in all datasets but BEIR and MIRACL; fusion typically matches or exceeds the better of the two single runs. Next, we look at the statistical significance of these results using pairwise permutation tests comparing BM25Q, RRF, and NAF to BoW.

As shown in Table 7, relative to BoW, BM25Q shows a small but statistically significant drop in nDCG@10 across all tasks ($\Delta = -0.8$, 95% CI $[-1.4, -0.2]$, $p = 0.005$) and an *NS* change in Recall@100 with $p = 0.408$. In contrast, fusing BoW and BM25Q is a low-effort win. Both RRF and NAF yield a clear recall gain with no nDCG@10 harm overall (Recall@100 $\Delta = +1.1$, $p < 0.001$; nDCG@10 *NS*).

Figure 3 visualizes these findings by showing, for each retrieval method, its average rank within each benchmark. For each task, the four retrieval methods are ranked from 1 (best) to 4 (worst) based on effectiveness. When methods tie, they are assigned the mean of the tied ranks (e.g., methods tied for ranks 2 and 3 are both assigned rank 2.5). These ranks are then averaged across all tasks in the benchmark. The figure shows that BoW and BM25Q alternate in rank depending on the evaluation metric, indicating comparable performance. In contrast, both fusion methods achieve consistently lower (i.e., better) average ranks across benchmarks, with NAF producing the strongest overall results.

6 Conclusion

This work establishes strong, reproducible baselines for reasoning-intensive retrieval on the BRIGHT benchmark by integrating lexical, dense, sparse, and reasoning-focused retrievers, along with listwise LLM rerankers, into widely used toolkits including Anserini, Pyserini, and RankLLM. These integrations lower the barrier to entry for researchers and provide a transparent foundation for evaluating retrieval pipelines in the emerging retrieval-augmented generation (RAG) setting.

Our analysis reveals several key findings. First, query-side BM25 (BM25Q), although under-documented in prior work, substantially improves lexical retrieval effectiveness on BRIGHT's long queries and is necessary to faithfully reproduce previously reported baselines. However, its benefits do not consistently generalize across standard IR benchmarks. In contrast, simple fusion of BoW and BM25Q—particularly normalized average fusion (NAF)—provides the most robust and generalizable configuration, yielding consistent recall improvements without harming early precision. These results suggest that modest modifications to classical lexical retrieval can remain highly effective even in reasoning-oriented settings.

Second, reasoning-focused retrievers significantly outperform generic dense and sparse retrievers, demonstrating the importance of task-aligned representation learning. Listwise LLM reranking further improves effectiveness, particularly for weaker first-stage retrievers, though its gains diminish as first-stage retrieval quality improves and may even degrade performance when reranker capability is insufficient relative to retriever strength. This highlights the importance of considering the interaction between retrieval and reranking components rather than optimizing them independently.

Finally, our audit of the BRIGHT corpus identifies data quality issues—including duplicate documents, empty passages, and missing relevance labels—that can affect evaluation outcomes. Addressing these issues improves the reliability of effectiveness measurements and underscores the critical role of dataset construction and preprocessing in reasoning-oriented retrieval benchmarks.

Taken together, our contributions provide both practical guidance and reproducible infrastructure for evaluating retrieval systems on reasoning-intensive tasks, and clarify which established retrieval practices remain effective and which require reconsideration in the era of LLM-driven retrieval.

7 Limitations and Future Work

Reasoning-Aware Reranking. While general-purpose LLMs as listwise rerankers provide substantial gains, specialized reasoning-aware pointwise rerankers trained with supervised fine-tuning or reinforcement learning achieve higher effectiveness on BRIGHT. Integrating such models into RankLLM and systematically studying their training strategies, scaling behavior, and robustness remains an important direction for closing the gap between reproducible references and leaderboard-level performance.

Query-Side and Agentic Retrieval Interventions. LLM-based query expansion, rewriting, and iterative retrieval have demonstrated strong potential for reasoning-intensive tasks. Future work should explore adaptive and multi-stage retrieval pipelines that dynamically refine queries based on intermediate results, while carefully evaluating the associated efficiency and latency trade-offs.

Hybrid and Adaptive Lexical–Neural Retrieval. Our findings show that simple fusion of lexical variants can provide robust gains, but the optimal combination likely depends on query characteristics, domain, and retriever strength. Future work should investigate adaptive fusion strategies, hybrid dense–sparse–lexical systems, and query-dependent weighting schemes that better leverage complementary retrieval signals.

Dataset Quality and Benchmark Reliability. Our corpus audit shows that preprocessing artifacts and incomplete relevance annotations can measurably affect evaluation. Future efforts should focus on improving dataset construction, including duplicate removal, passage segmentation quality, and more comprehensive relevance labeling, to ensure fair and reliable comparisons across systems.

Overall, advancing reasoning-oriented retrieval will require joint progress in retriever design, reranking methods, retrieval pipelines, and benchmark quality. We hope the reproducible references and analyses provided in this work serve as a foundation for continued research in this rapidly evolving area.

Acknowledgments

This research was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada. Additional funding was provided by Snowflake and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean Government (MSIT) (No. RS-2024-00457882, National AI Research Lab Project).

References

[1] Gianni Amati and Cornelis Joost Van Rijsbergen. 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 357–389.

[2] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv:1611.09268* (2016).

[3] Andrzej Białecki, Robert Muir, Grant Ingersoll, and Lucid Imagination. 2012. Apache Lucene 4. In *SIGIR 2012 Workshop on Open Source Information Retrieval*. sn, 17.

[4] Jianlyu Chen, Junwei Lan, Chaofan Li, Defu Lian, and Zheng Liu. 2025. ReasonEmbed: Enhanced text embeddings for reasoning-intensive document retrieval. *arXiv:2510.08252* (2025).

[5] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 Deep Learning Track. *arXiv:2102.07662* (2021).

[6] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. 2021. Overview of the TREC 2021 Deep Learning Track. In *Proceedings of the Thirtieth Text REtrieval Conference (TREC 2021)*.

[7] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2022. Overview of the TREC 2022 Deep Learning Track. In *Proceedings of the Thirty-First Text REtrieval Conference (TREC 2022)*.

[8] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 Deep Learning Track. *arXiv:2003.07820* (2020).

[9] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Hossein A. Rahmani, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2025. Overview of the TREC 2023 Deep Learning Track. *arXiv:2507.08890* (2025).

[10] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on RAG meeting LLMs: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 6491–6501.

[11] Revanth Gangi Reddy, JaeHyeok Doo, Yifei Xu, Md Arafat Sultan, Deevya Swain, Avirup Sil, and Heng Ji. 2024. FIRST: Faster Improved Listwise Reranking with Single Token Decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024)*. 8642–8652.

[12] Donna Harman. 1993. Overview of the first TREC conference. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*. 36–47.

[13] Donna Harman. 1995. Overview of the second Text REtrieval Conference (TREC-2). *Information Processing & Management* 31, 3 (1995), 271–289.

[14] Donna Harman. 1995. *Overview of the third Text REtrieval Conference (TREC-3)*. Vol. 225. DIANE Publishing.

[15] Chris Kamphuis, Arjen P. de Vries, Leonid Boytsov, and Jimmy Lin. 2020. Which BM25 Do You Mean? A Large-Scale Reproducibility Study of Scoring Variants. In *Advances in Information Retrieval: 42nd European Conference on IR Research (ECIR 2020), Part II* (Lisbon, Portugal). 28–34.

[16] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*. 6769–6781.

[17] Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. 2024. SPLADE-v3: New Baselines for SPLADE. *arXiv:2403.06789* (2024).

[18] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[19] Jimmy Lin. 2021. A Proposed Conceptual Framework for a Representational Approach to Information Retrieval. *arXiv:2110.01529* (2021).

[20] Jimmy Lin, Arthur Haonan Chen, Carlos Lassance, Xueguang Ma, Ronak Pradeep, Tommaso Teofili, Jasper Xian, Jheng-Hong Yang, Brayden Zhong, and Vincent Zhong. 2025. Gosling Grows Up: Retrieval with Learned Dense and Sparse Representations Using Anserini. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2025)* (Padua, Italy). 3223–3233.

[21] 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)*. 2356–2362.

[22] Jimmy Lin and W. John Wilbur. 2007. PubMed Related Articles: A Probabilistic Topic-Based Model for Content Similarity. *BMC Bioinformatics* 8, 1 (2007), 423.

[23] Wenhan Liu, Xinyu Ma, Weiwei Sun, Yutao Zhu, Yuchen Li, Dawei Yin, and Zhicheng Dou. 2025. ReasonRank: Empowering Passage Ranking with Strong Reasoning Ability. *arXiv:2508.07050* (2025).

[24] Meixiu Long, Duolin Sun, Dan Yang, Junjie Wang, Yue Shen, Jian Wang, Peng Wei, Jinjie Gu, and Jiahai Wang. 2025. DIVER: A Multi-Stage Approach for Reasoning-Intensive Information Retrieval. *arXiv:2508.07995* (2025).

[25] Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-Shot Listwise Document Reranking with a Large Language Model. *arXiv:2305.02156* (2023).

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

[27] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context Retrieval-Augmented Language Models. *Transactions of the Association for Computational Linguistics* 11 (2023), 1316–1331.

[28] Stephen Robertson, S. Walker, M. M. Hancock-Beaulieu, M. Gatford, and A. Payne. 1996. Okapi at TREC-4. In *Proceedings of the Fourth Text REtrieval Conference (TREC 4)*. 73–96.

[29] Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. 1995. Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference (TREC 3)*. 109–126.

[30] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (2009), 333–389.

[31] Anne Schuth, Floor Sietsma, Shimon Whiteson, and Maarten de Rijke. 2014. Optimizing Base Rankers Using Clicks. In *Advances in Information Retrieval*, Maarten de Rijke, Tom Kenter, Arjen P. de Vries, ChengXiang Zhai, Franciska de Jong, Kira Radinsky, and Katja Hofmann (Eds.). 75–87.

[32] 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, et al. 2025. ReasonIR: Training Retrievers for Reasoning Tasks. *arXiv:2504.20595* (2025).

[33] Sahel Sharifymoghaddam, Ronak Pradeep, Andre Slavescu, Ryan Nguyen, Andrew Xu, Zijian Chen, Yilin Zhang, Yidi Chen, Jasper Xian, and Jimmy Lin. 2025. RankLLM: A Python Package for Reranking with LLMs. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2025)* (Padua, Italy). 3681–3690.

[34] Nadia Athar Sheikh, Daniel Buades Marcos, Anne-Laure Jousse, Akintunde Oladipo, Olivier Rousseau, and Jimmy Lin. 2024. CURE: A Dataset for Clinical Understanding & Retrieval Evaluation. *arXiv:2412.06954* (2024).

[35] 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* (2025).

[36] 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 (EMNLP 2023)*. 14918–14937.

[37] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

[38] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-Pack: Packaged Resources to Advance General Chinese Embedding. *arXiv:2309.07597* (2023).

[39] Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the Use of Lucene for Information Retrieval Research. In *Proceedings of the 40th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017)*. Tokyo, Japan, 1253–1256.

[40] Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2023. MIRACL: A Multilingual Retrieval Dataset Covering 18 Diverse Languages. *Transactions of the Association for Computational Linguistics* 11 (09 2023), 1114–1131.