# Lightweight Embedded Hybrid Retrieval for Local LLMs: Unified Candidate Space and Deployment Trade-offs

1st Xize Han
School of Computer Science and Engineering
Hunan University of Information Technology
Changsha, China
hxz040823@163.com

2nd Xin Xie*
School of Computer Science and Engineering
Hunan University of Information Technology
Changsha, China
xiexin1@hnuit.edu.cn

***Abstract*—Local retrieval-augmented generation (RAG) must balance offline operation, resource use, and retrieval quality. When sparse and dense channels cover different entities, index coverage and rank-fusion effects are easily confounded. We build a unified candidate space of 1,696 educational entities and evaluate SQLite FTS5, BGE-M3 dense retrieval, and reciprocal rank fusion (RRF). Over 93 questions with judged relevant entities (186 query instances across two formulations), RRF reaches NDCG@5 of 0.8916 and MRR@5 of 0.9330, improving NDCG@5 over Dense and BM25 by 0.1124 and 0.1781. A fixed-list quota ablation shows how a deployment source policy changes top-five coverage and graded ranking. The results support evaluating index coverage, fusion, and deployment policy separately in local RAG.**

***CCS Concepts:* Information systems → Retrieval models and ranking; Evaluation of retrieval results; Computing methodologies → Natural language processing**

***Index Terms*—local RAG; hybrid retrieval; unified candidate space; reciprocal rank fusion; query-shell stripping; deployment trade-offs**

## I. Introduction

Retrieval-augmented generation (RAG) supplies large language models (LLMs) with retrievable external evidence [1, 2]. In a local teaching assistant, course materials, lecture notes, and user notes generally need to remain on the machine, while resource use and response time also matter. We therefore study local RAG retrieval that can be embedded in a desktop application, measuring retrieval and ranking cost separately from query-embedding cost.

Sparse lexical retrieval and dense vector retrieval use different matching signals and may complement one another [3, 4]. However, three factors can be confused in evaluations over a heterogeneous local corpus. First, the channels may index different entities. If BM25 covers only private user chunks while Dense also covers built-in knowledge nodes and questions, a direct comparison mixes candidate visibility with ranking ability. Second, templated natural-language questions may not fit a particular lexical query construction. Conversational wrappers, for example, can change the constraints of a SQLite FTS5 phrase query. We examine rule-based query-shell stripping and trigram construction in this implementation without generalizing the issue to all Chinese lexical retrieval. Third, a deployment source quota may conflict with global relevance order. RRF merges candidates by channel rank [5], while tie handling and source selection may further affect the top-K list; these mechanisms should be evaluated separately.

To address these issues, we align BM25 and Dense to the same set of 1,696 entities and analyze index coverage, rank fusion, and deployment policy separately. Our contributions are as follows:

(1) We design a unified candidate-space evaluation that compares BM25, Dense, and hybrid retrieval under identical entity coverage and separately reports the change accompanying index expansion, reducing coverage-related confounding in the ranking comparison.

(2) We construct an evaluation set with 100 base questions, two query formulations, and 847 pooled query–entity pairs. Two human raters and three model raters contribute relevance judgments, and paired bootstrap analysis resamples at the base-question level.

(3) We implement a local reference system using SQLite FTS5, BGE-M3 dense retrieval, RRF, and deterministic tie handling. We compare a unified-space evaluation with a deployment copy containing a source policy and report retrieval and embedding costs separately. Frozen rankings, relevance labels, and analysis code are available at https://github.com/xuzeTop1/RAG_RRF.

## II. Related Work

### A. Dense and Sparse Retrieval

Sparse methods such as BM25 rank by term matching and corpus statistics [4, 6], while dense methods use learned query and document representations, including DPR [7], Contriever [8], and BGE-M3 [9]. COIL stores contextualized

token representations in inverted lists to combine exact matching with contextual information [10]. ColBERT uses token-level late interaction to capture fine-grained query–document matches [11]. We use BM25 and single-vector BGE-M3 as directly comparable baselines because their retrieval stages cover the same 1,696 entities within the local implementation. COIL and ColBERT are better suited where per-token representations and a late-interaction scoring pass over the candidate set are affordable: both trade index size and per-query work for tighter coupling of lexical and contextual matching, which is complementary to the coverage-versus-fusion separation isolated here. They are kept out of the paired comparison rather than judged unsuitable, and remain candidate upgrades once the device budget allows. COIL and ColBERT offer different matching granularity; measuring their quality and indexing costs in this setting would require separate token-level indexes and resource accounting.

### *B. Hybrid Retrieval and Rank Fusion*

Hybrid retrieval combines complementary signals from multiple channels. Because BM25 scores and vector similarity have different directions and scales, directly weighting them usually requires normalization and a weight choice. Reciprocal rank fusion (RRF) instead uses each candidate's channel rank and does not compare raw scores directly [5]. Learned methods such as RankT5 can further optimize ranking [13], but add training or inference cost.

### *C. Query Reformulation and Understanding*

Query transformation reduces the gap between user wording and retrievable content. Contextual question rewriting resolves dialogue-dependent references [16]; LLM-based RAG rewriting can formulate a search query for a downstream retriever [17]; and Query2doc expands a query with a generated pseudo-document [18]. Our rule-based shell stripper addresses the narrower case of recognized surface wrappers: it is deterministic, runs locally without another model call, and leaves the lexical target available to FTS5. Open-ended paraphrases, synonym substitutions, and multi-turn references call for richer context-dependent rewriting. The two suffix templates in this benchmark test the local rule within its intended input pattern.

### *D. LLM-Assisted Relevance Evaluation*

LLMs have been used to evaluate text quality and retrieval relevance. Prior work examines agreement between model and human judgments, possible position, length, and self-preference biases [19–22], and the effect of model-generated relevance labels on retrieval evaluation [23]. Model-assisted labeling therefore still calls for human judgments, agreement analysis, and label-sensitivity checks.

This study evaluates a small local RAG retrieval layer under a common candidate space. SQLite FTS5, BGE-M3, and RRF form the reference system; the aligned comparisons separate index coverage from fusion, and the fixed-candidate ablation isolates the source quota. Human and model-assisted labels, bootstrap intervals, and staged timing records characterize quality and deployment cost within this setting.

## III. Problem Formulation

We study heterogeneous knowledge retrieval for local RAG, distinguishing ranking quality in a unified candidate space from multi-source results under a deployment configuration.

Let the full corpus consist of three mutually disjoint entity sets: knowledge nodes, questions, and private user chunks. Their union and disjointness are given in Eqs. (1) and (2):

$$\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3 \tag{1}$$

$$\mathcal{D}_i \cap \mathcal{D}_j = \emptyset, \quad i \neq j \tag{2}$$

For each entity d, the retrieval text $T_d$ is, respectively, “title + summary,” “title + question stem + answer,” or the private-chunk body, truncated to 8,000 characters. In production, BM25 covers 159 private chunks and Dense covers 1,537 built-in entities; the unified candidate-space evaluation aligns both methods to all 1,696 entities.

We address three research questions:

RQ1: After aligning index coverage, how does RRF ranking compare with the individual channels?

RQ2: Are RRF's relative gains stable under different relevance-label rules?

RQ3: What trade-off does a deployment configuration with a source quota show between retrieval coverage and early-rank quality?

We evaluate Hit@K, NDCG@5, and MRR@5. Here K is the evaluation cut-off and k is the RRF smoothing constant; by default, K = 5 and k = 60.

| Symbol | Meaning |
|---|---|
| D, \|D\| | Unified retrieval corpus (1,696 entities); subscripts 1, 2, and 3 denote knowledge nodes, questions, and private chunks. |
| q, σ(q) | Raw query and rule-stripped query. |
| $R_5$(q) | Top-five results for query q, with K = 5. |
| G(q) | Relevant-entity set for query q. |
| K | Evaluation cut-off / context capacity (K = 5). |
| k | RRF smoothing constant (k = 60). |
| F | Minimum opportunity reserved for each source (F = 2). |
| $L_s$, $L_d$ | Top-20 lists from sparse and dense channels. |

## IV. METHOD AND SYSTEM DESIGN

Figure 1 shows the local hybrid retrieval system and its two evaluation paths. Retrieval and storage are embedded in a Rust/Tauri desktop application, data reside in local SQLite, and a local Ollama service computes query vectors. Path A evaluates ranking and fusion in the unified candidate space without a source quota. Path B simulates asymmetric sparse-index coverage and a source quota in deployment. The evaluation copy also adds dense vectors for private chunks, so its Dense coverage differs from the production configuration. Python aligns entity keys for unified-space fusion; Section VI-E reports runtime cost.

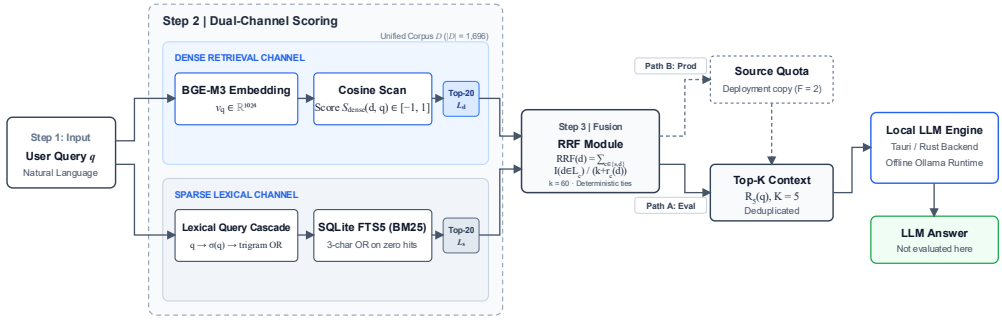



Figure 1. Local embedded hybrid retrieval system and evaluation paths. Path A is the unified candidate-space evaluation; Path B is the deployment copy with a source quota (F = 2). The evaluation copy additionally embeds private chunks.

### A. Candidate Space and Local Indexes

The corpus contains 773 knowledge nodes, 764 questions, and 159 private document chunks: 1,696 entities in total. Their retrieval texts are, respectively, "title + summary," "title + question stem + answer," and the chunk body, truncated to 8,000 characters. A global entity key connects each entity's representations across indexes.

In production, BM25 indexes only 159 private chunks, while Dense covers 1,537 built-in entities. To control differences in candidate visibility, the evaluation copy adds BGE-M3 vectors for the 159 private chunks and creates an FTS5 index for all entities in the unified-space experiment. Denoting the sparse and dense coverage sets by $D_s$ and $D_d$, both channels can retrieve the same set, as in Eq. (3):

$$\mathcal{D}_s = \mathcal{D}_d = \mathcal{D}, \quad |\mathcal{D}| = 1{,}696 \tag{3}$$

The unified space thus compares ranking and fusion under fixed entity coverage. The deployment copy is used to study the joint effects of asymmetric sparse indexing and source policy.

### B. Sparse Lexical Retrieval

The sparse channel uses SQLite FTS5 with its trigram tokenizer. For templated questions, a deterministic query-shell stripping operator σ(q) repeatedly removes predefined question prefixes, suffixes, and surrounding punctuation until the query stops changing. It removes only wrappers listed in the dictionary and does not perform semantic rewriting.

FTS5 uses a three-stage cascade triggered by zero hits:

Construct a lexical query from the raw query q.

If it returns no results, use the stripped query σ(q).

If it still returns no results, split the compact form of σ(q) (or of q if σ(q) has fewer than three characters) into overlapping three-character windows joined by OR, using at most 32 windows.

Queries shorter than three Unicode characters cannot be matched directly by the FTS5 trigram tokenizer and therefore rely only on the dense channel [12]. Candidates are sorted in ascending order of SQLite FTS5's bm25() value ($k_1$ = 1.2, b = 0.75); the top 20 form $L_s$.

In this evaluation set, only two suffixes occur, equivalent to "what does it mean" and "how is it calculated"; no predefined prefixes occur. These experiments therefore cannot evaluate the complete rule dictionary.

### C. Dense Semantic Retrieval

The dense channel encodes the raw query q without shell stripping. BGE-M3 [9] produces a 1,024-dimensional query vector $v_q$; its cosine similarity with an entity vector $v_d$ is defined in Eq. (4):

$$S_{\text{dense}}(d, q) = \frac{\boldsymbol{v}_q \cdot \boldsymbol{v}_d}{\|\boldsymbol{v}_q\|_2 \|\boldsymbol{v}_d\|_2} \quad (4)$$

At this experimental scale of 1,696 entities, the system performs a full in-memory scan and returns the top 20 candidates $L_d$. This implementation is suitable for the small local corpus studied here; its runtime should not be extrapolated to large-scale vector retrieval.

### D. RRF Fusion and Deterministic Ordering

Given sparse and dense candidate lists $L_s$ and $L_d$, the RRF score of entity d is defined in Eq. (5):

$$\text{RRF}(d) = \sum_{c \in \{s,d\}} \frac{I(d \in L_c)}{k + r_c(d)} \quad (5)$$

Here $r_c$(d) is the rank of d in channel c, and k is the smoothing constant, set to 60 by default [5]. We also examine k = 1.

Before fusion, mirrored lexical keys are mapped to global entity keys and duplicate entities are removed. Equal RRF scores are ordered by two-channel presence; for the best channel rank, odd ranks favor sparse and even ranks favor dense, followed by source and global entity ID. In conditional_93, the archived prefix-key variant changes NDCG@5 by 0.0015319 for template, 0 for bare, and 0.0007659 when the two forms are pooled; MRR@5 is unchanged. The rule provides deterministic ranking rather than a substantial effectiveness gain.

### E. Source Quota and Deployment Path

The deployment configuration adds a source quota to give private material a minimum opportunity for display. The policy does not change relevance scores; it selects results after RRF ranking.

Candidates are divided into private chunks and nonprivate entities. When both sources have candidates and their total exceeds K, up to F = 2 results from each source are reserved in their original fusion order. Remaining slots are filled by RRF order, and selected entities retain their relative order in the original fused list. If there are too few candidates or only one source, the fused results are simply truncated.

The unified-space main experiment, rrf_k60_unified, applies no source quota and estimates the rank-fusion effect. The deployment copy, hybrid_prod_quota, retains BM25's private-chunk-only index and adds a source quota. Its Dense channel has added private-chunk vectors and is therefore not identical to the production configuration without those vectors. A fixed-candidate ablation isolates the quota's effect in Section VI-A.

## V. Experimental Setup

### A. Evaluation Data and Query Construction

We construct a local educational knowledge-retrieval evaluation set with 100 base questions spanning computer science, higher mathematics and probability, university physics, education, and introductory law. It is designed to analyze retrieval over a multidisciplinary educational corpus, not to represent the distribution of real user queries.

Each base question has two formulations: template ($q_T$) is a full question with a question wrapper, and bare ($q_B$) is the target concept alone. For 68 of the 100 questions, template adds one of two Chinese suffixes to the target term, equivalent to “what does it mean” or “how is it calculated”; for the other 32, template and bare are identical. The relation in this dataset is therefore given by Eq. (6):

$$\sigma(q_T) = q_B \quad (6)$$

This setup tests known suffix wrappers but cannot establish robustness to prefixes, expressions outside the dictionary, or naturally open-ended queries.

Both formulations share a base-question ID. In the main analysis, 93 questions with at least one relevant entity in the judged pool yield 186 query instances. Statistical inference still treats the 93 base questions as independent resampling units.

### B. Candidate Pool and Relevance Annotation

Using pooling [24], we collect the top five results of BM25, Dense, and Hybrid for template and bare queries. Deduplication by query–entity pair yields 847 candidate pairs. Entities outside the pool are unjudged and treated as nonrelevant in metric calculations; the results therefore depend on pool depth and participating systems. The unified-space BM25 and fused runs were not pooled: 22% of unified BM25's top-five results are unjudged, versus 5–8% for the fused runs and none for Dense (Section VI-D). The pool was assembled from an earlier run of these six configurations; the frozen runs released with this paper reproduce the per-question pool exactly for 44 of the 93 questions, so the pool cannot be re-cut from them.

Two human raters and three model raters independently make binary relevance judgments for each pair. The human raters (H1, H2) both hold a bachelor's degree in computing: H1 is one of the authors, and H2 is an unpaid personal contact recruited for the task. Both labeled blind through a web interface; the model raters are Qwen 3.8 Flash (A), GPT 5.6 (B),

and DeepSeek V4.1 (C), each with a recorded temperature of 0.0. A pilot model rater, Gemini 3.8 Flash (D), was excluded under the protocol's redundancy rule (raw agreement above 95% with a panel member): it agreed with A on 833 of 847 pairs (98.35%). All five votes have equal weight; at least three votes mark a pair relevant. A post-freeze review of five borderline questions (q013, q018, q055, q072, q077) then adjusts the frozen label set G(q) used in the main text. It was drafted with an LLM reviewer after the rater exports were frozen and confirmed item by item by the authors, and the reproduction package keeps the archive field names ai_review_supplement and ai_review_correction. These adjustments are not votes: on q055 three entities the majority had marked relevant were removed, and on q018, q072 and q077 one unjudged entity each was added, its candidate having fallen outside the pooling depth. Section VI-F repeats the headline comparison without them. This is a human-and-model reference standard, not a purely human gold standard.

Overall Fleiss' κ among the five raters is 0.5774 [25], with unanimous judgments on 552 of 847 pairs (65.17%). The two human raters agree on 618 pairs (72.96%), giving Cohen's κ of 0.4207; 615 of their shared judgments match the final labels (Table I). We also compare alternative label rules and rater panels in Section VI-F.

TABLE I. RATER AGREEMENT (H1, H2: HUMAN RATERS; A–C: MODEL RATERS)

| Rater pair | Cohen's κ | Raw agreement |
|---|---|---|
| H1–H2 | 0.4207 | 72.96% |
| H1–A | 0.6622 | 83.71% |
| H1–B | 0.6777 | 84.53% |
| H1–C | 0.3772 | 71.07% |
| H2–A | 0.5460 | 81.94% |
| H2–B | 0.6188 | 85.36% |
| H2–C | 0.7757 | 93.39% |
| A–B | 0.7334 | 88.31% |
| A–C | 0.5137 | 80.99% |
| B–C | 0.5605 | 83.47% |
| All five raters | 0.5774* | 552/847 65.17% |

* Fleiss' κ is used for all five raters.

Seven of the 100 base questions have no majority-vote relevant entity in the current judged pool. We report two scopes: conditional_93 includes only the 93 questions with a nonempty relevant set, whereas overall_100 includes all 100 and assigns zero retrieval metrics to those seven questions. The latter describes performance under the current pooled labels and does not evaluate no-answer detection.

### C. *Metrics and Statistical Analysis*

Results are first deduplicated by global entity key. Metrics use the binary relevant-entity set G(q):

Hit@K is the fraction of queries with at least one relevant entity in the top K, for $K \in \{1, 3, 5\}$.

MRR@5 is the mean reciprocal rank of the first relevant entity in the top five, with zero if none appears.

NDCG@5 is normalized discounted cumulative gain at five based on binary relevance.

We report template and bare separately and pool them with equal weight (both). Questions with $G(q) = \emptyset$ are excluded from conditional_93 and scored zero in overall_100.

Paired bootstrap resamples 93 base-question IDs 2,000 times, preserving both formulations and system pairing (actual seed 20260933 = base seed 20260916 + 17). We report 95% percentile intervals [26] without multiple-comparison correction; they support individual comparisons, not family-wise significance.

## VI. RESULTS AND ANALYSIS

### A. *Coverage and Early-Rank Quality in the Deployment Copy*

Table II compares BM25, Dense, and Hybrid in the evaluation deployment copy. Dense covers all 1,696 entities because private-chunk vectors were added; BM25 still covers only 159 private chunks, and Hybrid fuses both channels.

TABLE II. RESULTS IN THE DEPLOYMENT COPY (CONDITIONAL 93, TEMPLATE; S/D: SPARSE/DENSE COVERAGE; HYBRID APPLIES THE QUOTA AT RETRIEVAL DEPTH 20)

| Metric | Hybrid | BM25 | Dense |
|---|---|---|---|
| Candidate space | 159 S + 1,696 D | 159 S | 1,696 D |
| Hit@1 | 0.7097 | 0.3011 | 0.7957 |
| Hit@3 | 0.9140 | 0.3333 | 0.8817 |
| Hit@5 | 0.9462 | 0.3333 | 0.8925 |
| MRR@5 | 0.8116 | 0.3172 | 0.8342 |
| NDCG@5 | 0.7725 | 0.2343 | 0.7708 |

Compared with Dense, Hybrid raises Hit@5 from 0.8925 to 0.9462, but lowers MRR@5 from 0.8342 to 0.8116; NDCG@5 is nearly unchanged. Thus top-five coverage improves without a corresponding improvement in the position of the first relevant result.

The quota ablation separates fusion from source selection (Figure 2). It pools template and bare queries, unlike the template-only scope of Table II. At retrieval depth 20, switching the quota changes none of 186 lists, so the Table II differences mainly reflect channel fusion. At the actual five-slot output, the quota changes 103 of 186 lists: Hit@5 rises from 0.9462 to 0.9570, while NDCG@5 falls from 0.7759 to 0.7544. Paired bootstrap (2,000 resamples; actual seed 20260929) gives ΔNDCG@5 (quota off minus on) of +0.0215, with 95% CI [0.0025, 0.0402]; the Hit@5 and MRR@5 intervals cross zero.

The deployment copy's main coverage–ranking trade-off therefore comes from channel fusion. The quota lowers NDCG@5 at five returned results; evidence for its effects on Hit@5 and MRR@5 remains inconclusive.

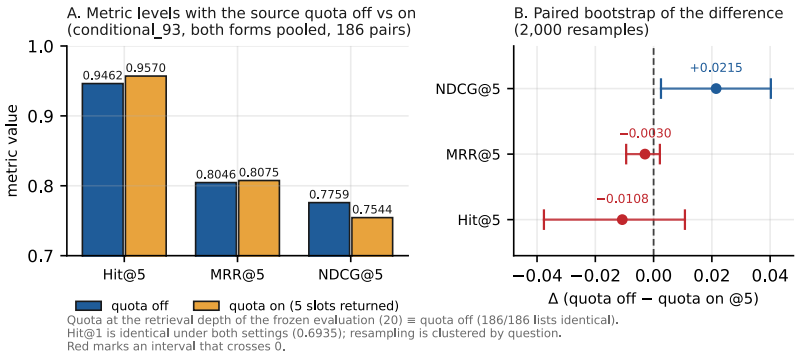



Figure 2. Ablation of the source quota

## *B. Main Results in the Unified Candidate Space*

Table III compares retrieval when BM25 and Dense each cover all 1,696 entities and no source quota is applied. Interleave is a simple alternating-fusion baseline.

TABLE III. UNIFIED CANDIDATE-SPACE RESULTS (HIT@K, MRR@5, NDCG@5)

| Strategy | H@1 | H@3 | H@5 | MRR | NDCG |
|---|---|---|---|---|---|
| **conditional_93 template** | | | | | |
| RRF k60 | 0.9140 | 0.9677 | 0.9892 | 0.9457 | 0.9016 |
| RRF k1 | 0.9140 | 0.9570 | 0.9892 | 0.9407 | 0.8916 |
| Interleave | 0.8925 | 0.9570 | 0.9785 | 0.9278 | 0.8751 |
| Dense | 0.7957 | 0.8817 | 0.8925 | 0.8342 | 0.7708 |
| BM25 | 0.8602 | 0.8710 | 0.9032 | 0.8726 | 0.7135 |
| **conditional_93 bare** | | | | | |
| RRF k60 | 0.8710 | 0.9677 | 0.9785 | 0.9203 | 0.8816 |
| Dense | 0.7742 | 0.8602 | 0.8925 | 0.8211 | 0.7875 |
| BM25 | 0.8602 | 0.8710 | 0.9032 | 0.8726 | 0.7135 |
| **conditional_93 both** | | | | | |
| RRF k60 | 0.8925 | 0.9677 | 0.9839 | 0.9330 | 0.8916 |
| Dense | 0.7849 | 0.8710 | 0.8925 | 0.8277 | 0.7792 |
| BM25 | 0.8602 | 0.8710 | 0.9032 | 0.8726 | 0.7135 |
| **overall_100 both** | | | | | |
| RRF k60 | 0.8300 | 0.9000 | 0.9150 | 0.8677 | 0.8292 |
| Dense | 0.7300 | 0.8100 | 0.8300 | 0.7697 | 0.7246 |
| BM25 | 0.8000 | 0.8100 | 0.8400 | 0.8115 | 0.6636 |

Under conditional_93/both, RRF (k = 60) reaches MRR@5 of 0.9330 and NDCG@5 of 0.8916, exceeding Dense and BM25 in NDCG@5 by 0.1124 and 0.1781. For template queries, RRF reaches NDCG@5 of 0.9016, improvements of 0.1308 and 0.1881 over Dense and BM25; the differences are also positive for bare queries. Absolute values fall under overall_100, while the general pattern remains.

For template queries, k = 60 and k = 1 yield NDCG@5 values of 0.9016 and 0.8916. A smaller k accentuates rank differences near the top of each channel, while a larger k compresses within-list rank gaps and gives shared presence across channels relatively more weight. We retain the conventional k = 60 setting [5] and use k = 1 as a top-heavy contrast. Re-fusing the same two channel lists at intermediate constants leaves the result flat: pooled over both formulations,

NDCG@5 stays between 0.8915 and 0.8918 for k = 10, 20, 40, and 60, and MRR@5 between 0.9325 and 0.9330, while k = 1 falls to 0.8805 and 0.9262. The advantage over Dense holds at every k tested, from +0.1013 [+0.0433, +0.1632] at k = 1 to +0.1124 [+0.0555, +0.1756] at k = 60 (question-clustered percentile bootstrap, 2,000 draws). The conventional k = 60 therefore sits on a plateau rather than at a tuned optimum: within this corpus and these two query formulations only the extreme top-heavy setting costs accuracy, so k may be chosen for other reasons than expected ranking gain.

### *C. Index Expansion and Fusion under Fixed Coverage*

Index expansion and rank fusion represent different system changes. When BM25 expands from 159 private chunks to 1,696 entities, template-query NDCG@5 rises from 0.2343 to 0.7135, a difference of 0.4792. Because this also changes BM25's corpus statistics, the difference describes a change in the overall configuration, not a pure coverage effect.

Within the unified candidate space, BM25, Dense, and RRF obtain NDCG@5 values of 0.7135, 0.7708, and 0.9016, respectively. RRF exceeds the two individual channels by 0.1881 and 0.1308. Since they can retrieve the same entities, this comparison more directly reflects differences in fused ranking.

We therefore report index expansion and fusion under fixed coverage separately (Figure 3). Their differences are neither independent causal effects nor additive contributions to performance.

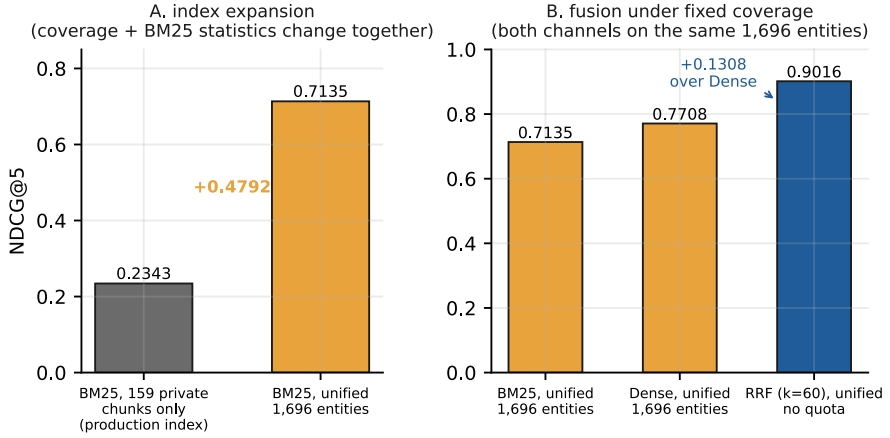



Figure 3. Index expansion and fusion under fixed coverage (conditional_93, template)

### *D. Paired Bootstrap Uncertainty Analysis*

Table IV reports metric differences and 95% percentile intervals from 2,000 paired bootstrap resamples at the base-question ID level.

TABLE IV. PAIRED BOOTSTRAP DIFFERENCES AND 95% CONFIDENCE INTERVALS

| Comparison | Form | Metric | Δ | 95% CI |
|---|---|---|---|---|
| RRF–BM25 | T | NDCG | +0.1881 | [+0.1383,+0.2398] |
| RRF–BM25 | T | MRR | +0.0731 | [+0.0312,+0.1210] |
| RRF–BM25 | B | NDCG | +0.1681 | [+0.1246,+0.2118] |
| RRF–BM25 | B | MRR | +0.0477 | [+0.0172,+0.0833] |
| RRF–Dense | T | NDCG | +0.1308 | [+0.0698,+0.1961] |
| RRF–Dense | T | MRR | +0.1115 | [+0.0457,+0.1794] |
| RRF–Dense | B | NDCG | +0.0940 | [+0.0363,+0.1571] |
| RRF–Dense | B | MRR | +0.0991 | [+0.0369,+0.1659] |
| BM25 U–P | both | NDCG | +0.4792 | [+0.3840,+0.5708] |
| BM25 U–P | both | MRR | +0.5554 | [+0.4446,+0.6624] |

T and B denote template and bare; U and P denote the unified and production indexes. All eight within-unified-space intervals comparing RRF with BM25 or Dense are above zero, as are the two index-expansion comparisons. Without multiple-comparison correction, these intervals support individual comparisons, not control of the family-wise error rate. Re-scoring with unjudged entities dropped (condensed lists), or counted as relevant within the reported Top-5 window with one relevance set shared per question, keeps all NDCG@5 and RRF–Dense MRR@5 intervals above zero; the RRF–BM25 MRR@5 comparison then crosses or touches zero in three of its four intervals (the exception, +0.0591 [+0.0188, +0.1075], is the condensed templated formulation). Figure 4 plots the Table IV intervals.

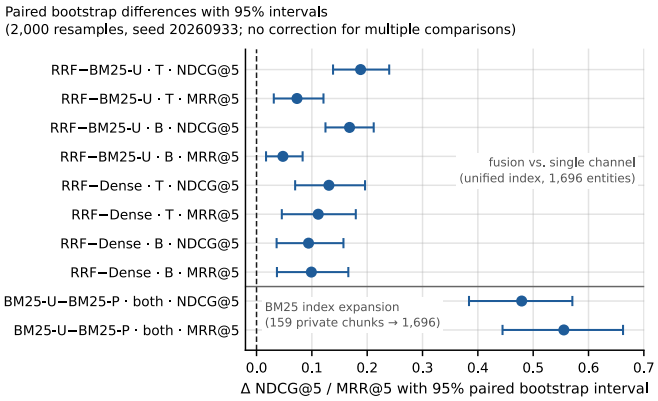



Figure 4. Paired bootstrap differences and 95% percentile intervals (2,000 resamples; actual seed 20260933; no multiple-comparison correction). T = template, B = bare, U = unified index, and P = production index.

### *E. Query-Preprocessing Diagnostics and Runtime Cost*

Development traces show that a strict FTS5 phrase built from a wrapped question can return no lexical hit. The cascade then tries the stripped form and a trigram OR fallback (Section IV-B), illustrating how known wrappers interact with the local index. These traces use different index and label snapshots, so we use them as implementation diagnostics. In the main benchmark, Dense obtains NDCG@5 of 0.7708 for template and 0.7875 for bare queries.

On an Intel i5-13500HX under Windows 11, the native Rust path was timed over 100 runs with a preloaded query vector: FTS5 lookup, a cosine scan of 1,696 entities, RRF fusion, and deterministic sorting averaged 12.662 ms (sample sd 1.114 ms, P95 13.801 ms, maximum 19.324 ms). The individual runs and evaluation-copy identifier were not retained. For inspectable per-query timing, the released ranking files provide one elapsedMicros value per question and configuration: Hybrid averages 12.969/12.778 ms for template/bare on the deployment copy and 16.768/16.112 ms on the unified copy. These single-pass records and the 100-run summary represent different measurements. Local CPU query embedding with BGE-M3 averaged 174.3 ms for template and 41.0 ms for bare; embedding, answer generation, scheduling, and interface rendering are outside the 12.662 ms retrieval figure. The unified evaluation database occupies 21.70 MiB, and the raw payload of 1,696 float32 vectors of dimension 1,024 is approximately 6.62 MiB.

### *F. Relevance-Label Sensitivity*

To test dependence on the main label definition, we hold retrieval results fixed and change only the relevance-label rule. The number of eligible questions can differ across rules, so we compare methods within each rule rather than comparing absolute metrics across rules. Table V reports the number with usable frozen rankings, not necessarily the full eligible set under each alternate rule. Under permissive human labels, the true nonempty conditional set has 99 questions; q074, q087, q092, q095, q099, and q100 lack frozen rankings. Under the six-rater ≥3/6 rule, the true set has 96 questions; q074, q095, and q099 lack frozen rankings. Both rows therefore report metrics only for the fixed 93-question evaluation set. Until rankings are obtained for the seven previously excluded questions, we cannot establish whether conclusions for either rule would persist or reverse on its true conditional set.

TABLE V. SENSITIVITY TO RELEVANCE-LABEL RULES (†: NDCG@5 AVERAGED EQUALLY OVER THE TWO QUERY FORMULATIONS)

| Label rule | N | RRF† | Dense† | BM25† | Δ(RRF−Dense)† | Δ(RRF−BM25)† | 95% CI, RRF−Dense (bare) |
|---|---|---|---|---|---|---|---|
| Paper labels v3_frozen (with the post-freeze review) | 93 | 0.8916 | 0.7792 | 0.7135 | +0.1124 | +0.1781 | [+0.0363, +0.1571] |
| Pure five-rater majority (≥3/5; no review adjustments) | 93 | 0.8867 | 0.7879 | 0.7043 | +0.0987 | +0.1824 | [+0.0257, +0.1401] |
| Conservative human: both raters judge relevant | 84 | 0.8938 | 0.7962 | 0.7940 | +0.0976 | +0.0999 | [+0.0281, +0.1349] |
| Permissive human: either rater judges relevant | 93 | 0.8599 | 0.8076 | 0.5709 | +0.0523 | +0.2891 | [−0.0045, +0.0780] (crosses 0) |
| Model-only majority (three raters) | 93 | 0.8890 | 0.7873 | 0.7032 | +0.1018 | +0.1858 | [+0.0300, +0.1458] |
| Including excluded rater D | 93 | 0.8775 | 0.7987 | 0.6787 | +0.0788 | +0.1988 | [+0.0074, +0.1132] |

TABLE V. SENSITIVITY TO RELEVANCE-LABEL RULES (†: NDCG@5 AVERAGED EQUALLY OVER THE TWO QUERY FORMULATIONS)

| Label rule | N | RRF† | Dense† | BM25† | Δ(RRF−Dense)† | Δ(RRF−BM25)† | 95% CI, RRF−Dense (bare) |
|---|---|---|---|---|---|---|---|
| (≥3/6) | | | | | | | |
| Including excluded rater D (≥4/6) | 93 | 0.8852 | 0.7785 | 0.7070 | +0.1067 | +0.1782 | [+0.0327, +0.1490] |

RRF's gain over both individual channels remains positive under every label rule (Figure 5). Under conservative human labels, the relevant confidence intervals still exclude zero. Under permissive human labels, however, RRF minus Dense for bare queries is +0.0329 (95% CI [−0.0045, 0.0780]); because this interval includes zero, a positive difference is not established for that comparison. Including the previously excluded model rater changes effect sizes but not their direction.

These results suggest that RRF's overall advantage over the individual channels is fairly robust to most label definitions, although its magnitude depends on relevance rules and rater-panel composition. The relative gap between Dense and BM25 is especially sensitive to the labels. Every sensitivity analysis reuses the same top-five pooled candidate set, so none removes the limitations of pool depth or templated queries.

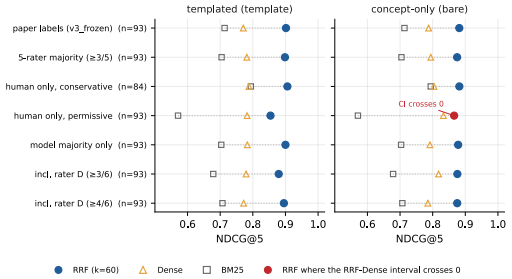



Figure 5. Effect of relevance-label definitions on the main finding. For the permissive-human and six-rater ⩾3/6 rows, n = 93 denotes the fixed sample with frozen rankings; the full nonempty sets contain 99 and 96 questions, respectively.

## VII. DISCUSSION, LIMITATIONS, AND ETHICS

### A. *Implications for Local RAG Evaluation*

Evaluation of local hybrid retrieval should distinguish index coverage, rank fusion, and deployment policy. A unified candidate space reduces differences in entity visibility and makes fusion comparisons easier to interpret. The source quota is a separate result-selection mechanism and should be reported separately from retrieval ranking. In deployment, greater coverage need not improve the top positions; neither Hit nor NDCG alone describes the full behavior.

### B. *Threats to Validity and Scope*

The benchmark captures course-concept queries with two observed suffix templates and binary pooled labels. Shell stripping targets those known wrappers; open-ended or multi-turn requests can instead call for the contextual or LLM-based rewriting discussed in Section II-C. Index expansion also changes BM25 statistics, and deployment comparisons combine several factors; the fixed-list ablation isolates the quota component. Bootstrap intervals are item-wise and unadjusted for multiple comparisons. Under permissive human labels, the bare-query RRF–Dense interval crosses zero. The reported effectiveness and runtime measurements concern this 1,696-entity educational corpus and one desktop CPU.

### C. *Annotation Transparency, Privacy, and Ethics*

Two human and three model raters formed the reference labels. H1, an author, supplied 378 positive judgments across the 847 common pairs; H2, recruited informally and without payment, supplied 161. Of 229 human disagreements, 223 were positive only for H1 and six only for H2. The author thus used a more inclusive relevance threshold in this pool, although these counts alone cannot determine which judgments are correct or the net error of the five-rater majority. Human–human Cohen's κ is 0.4207 with 72.96% raw agreement (Table I); Section VI-F examines alternative label rules. Both human raters consented to release of pseudonymized votes, and no formal ethics protocol was used. Some model annotation used cloud services; the later five-question review used an LLM draft confirmed by the authors and is recorded separately from blind panel votes.

## VIII. CONCLUSION

We built a unified candidate space with 1,696 entities for local educational RAG and compared BM25, Dense, and RRF hybrid retrieval. On 93 questions with a relevant entity in the judged pool (186 query instances), RRF (k = 60) without a source quota reaches NDCG@5 of 0.8916 and MRR@5 of 0.9330. Its NDCG@5 exceeds Dense and BM25 by 0.1124 and 0.1781; the main individual bootstrap intervals are above zero, without multiple-comparison correction.

Further analyses show distinct roles for index expansion, rank fusion, and source quota. RRF retains an advantage within the unified candidate space. In the evaluation deployment copy, Hybrid increases top-five coverage without

improving the rank of the first relevant result. At an actual five-slot output depth, the quota lowers NDCG@5, while the Hit@5 and MRR@5 difference intervals cross zero. The archived per-query timings distinguish retrieval cost on the deployment and unified copies.

The unified candidate-space design provides a practical way to measure rank-fusion gains separately from index coverage and source-selection policies in local RAG.

## Data and Code Availability

The reproduction package is available at https://github.com/xuzeTop1/RAG_RRF under the MIT License. It contains pseudonymized relevance labels and per-rater votes, frozen Top-20 rankings, aggregate results, analysis scripts, and reference retrieval code. From these frozen inputs, readers can recompute ranking metrics, bootstrap intervals, the quota and tie-key comparisons, the smoothing-constant sweep behind Section VI-B, and label-sensitivity analyses. The source corpus and retrieval index files are held separately; this package supports replay of the reported analyses rather than regeneration of corpus retrieval or the original annotation session.

## References


[1] P. Lewis, E. Perez, A. Piktus, et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 9459–9474, 2020.

[2] Y. Gao, Y. Xiong, X. Wang, et al., "Retrieval-augmented generation for large language models: A survey," arXiv preprint arXiv:2312.10997, 2023.

[3] N. Thakur, N. Reimers, A. Rücklé, et al., "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), 2021.

[4] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," Foundations and Trends® in Information Retrieval, vol. 3, no. 4, pp. 333–389, 2009. DOI: 10.1561/1500000019.

[5] G. V. Cormack, C. L. Clarke, and S. Büttcher, "Reciprocal rank fusion outperforms Condorcet and individual rank learning methods," in Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 758–759, 2009. DOI: 10.1145/1571941.1572114.

[6] S. E. Robertson, S. Walker, S. Jones, et al., "Okapi at TREC-3," in Overview of the Third Text REtrieval Conference (TREC-3), pp. 109–126, 1995.

[7] V. Karpukhin, B. Oğuz, S. Min, et al., "Dense passage retrieval for open-domain question answering," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6769–6781, 2020. DOI: 10.18653/v1/2020.emnlp-main.550.

[8] G. Izacard, M. Caron, L. Hosseini, et al., "Unsupervised dense information retrieval with Contriever," Transactions on Machine Learning Research (TMLR), 2022.

[9] J. Chen, S. Xiao, P. Zhang, et al., "BGE M3-Embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation," arXiv preprint arXiv:2402.03216, 2024.

[10] L. Gao, Z. Dai, and J. Callan, "COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List," in Proceedings of NAACL-HLT, pp. 3030–3042, 2021. DOI: 10.18653/v1/2021.naacl-main.241.

[11] O. Khattab and M. Zaharia, "ColBERT: Efficient and effective passage search via contextualized late interaction over BERT," in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 39–48, 2020. DOI: 10.1145/3397271.3401075.

[12] SQLite Consortium, "SQLite FTS5 Extension," SQLite Documentation. https://www.sqlite.org/fts5.html (accessed 2026-09-22).

[13] H. Zhuang, Z. Qin, R. Jagerman, et al., "RankT5: Fine-tuning T5 for text ranking with ranking losses," in Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), pp. 2308–2313, 2023. DOI: 10.1145/3539618.3592047.

[14] J. J. Rocchio, "Relevance feedback in information retrieval," in The SMART Retrieval System: Experiments in Automatic Document Processing, G. Salton, Ed., Englewood Cliffs, NJ: Prentice-Hall, pp. 313–323, 1971.

[15] H. K. Azad and A. Deepak, "Query expansion techniques for information retrieval: A survey," Information Processing & Management, vol. 56, no. 5, pp. 1698–1735, 2019. DOI: 10.1016/j.ipm.2019.05.009.

[16] A. Elgohary, D. Peskov, and J. Boyd-Graber, "Can you unpack that? Learning to rewrite questions-in-context," in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 5918–5924, 2019. DOI: 10.18653/v1/D19-1605.

[17] X. Ma, Y. Gong, P. He, H. Zhao, and N. Duan, "Query rewriting in retrieval-augmented large language models," in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 5303–5315, 2023. DOI: 10.18653/v1/2023.emnlp-main.322.

[18] L. Wang, N. Yang, and F. Wei, "Query2doc: Query expansion with large language models," in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 9414–9423, 2023. DOI: 10.18653/v1/2023.emnlp-main.585.

[19] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica, "Judging LLM-as-a-judge with MT-Bench and Chatbot Arena," in Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track, 2023. arXiv:2306.05685.

[20] J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai, C. Xu, W. Li, Y. Shen, S. Ma, H. Liu, Y. Wang, and J. Guo, "A survey on LLM-as-a-judge," arXiv preprint arXiv:2411.15594, 2024.

[21] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, "G-Eval: NLG evaluation using GPT-4 with better human alignment," in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2511–2522, 2023. DOI: 10.18653/v1/2023.emnlp-main.153.

[22] P. Wang, L. Li, L. Chen, Z. Cai, D. Zhu, B. Lin, Y. Cao, L. Kong, Q. Liu, T. Liu, and Z. Sui, "Large language models are not fair evaluators," in Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024), pp. 9440–9450, 2024. DOI: 10.18653/v1/2024.acl-long.511.

[23] H. A. Rahmani, C. Siro, M. Aliannejadi, N. Craswell, C. L. A. Clarke, G. Faggioli, B. Mitra, P. Thomas, and E. Yilmaz, "Judging the judges: A collection of LLM-generated relevance judgements," arXiv preprint arXiv:2502.13908, 2025.

[24] E. Voorhees and D. Harman, TREC: Experiment and Evaluation in Information Retrieval, MIT Press, 2005.

[25] J. Fleiss, "Measuring nominal scale agreement among many raters," Psychological Bulletin, vol. 76, no. 5, pp. 378–382, 1971. DOI: 10.1037/h0031619.

[26] B. Efron and R. J. Tibshirani, An Introduction to the Bootstrap, CRC Press, 1994.