

Hybrid Retrieval-Augmented Generation for Quantitative Aptitude Tutoring

Botsa Yaswanth Kumar¹, Prof. Dhavala Lalitha Bhaskari²

¹*M. Tech Student, Department of Computer Science and Systems Engineering, Andhra University College of Engineering, Visakhapatnam, India*

²*Professor, Department of Computer Science and Systems Engineering, Andhra University College of Engineering, Visakhapatnam, India*

Abstract—LLM chatbots deployed as tutors risk hallucinating numerical facts and formulas, which is especially damaging in quantitative-reasoning subjects. Retrieval-Augmented Generation (RAG) mitigates this by grounding generation in retrieved context, but most educational RAG systems fix a single retrieval strategy without justifying that choice empirically. This project implements and compares three retrieval configurations BM25 (lexical), dense (semantic), and hybrid (query-fusion of both) within a quantitative-aptitude tutoring chatbot spanning sixteen topics (Percentage, Interest, Ratio-Average, Time-Work, Time-Speed-Distance, Permutation-Probability, Number System, Mensuration, Ages-Alligation, Profit-Loss, Data Interpretation, Simplification-Approximation, LCM-HCF, Probability, Boats-Streams, Pipes-Cisterns). All configurations share an identical corpus, chunking, cross-encoder reranker, and generator LLM, isolating the retrieval strategy as the sole independent variable. On a 1000-question benchmark, BM25 and dense retrieval attain marginally higher precision, recall, and context relevancy, while hybrid retrieval achieves the highest answer relevancy (87.74%) at nearly 4.5× the latency of BM25 (3.29 s vs. 0.73 s). BM25 attains the best exact-match accuracy (91.00%) and ROUGE-L (75.96%) at the lowest latency. These results indicate that for narrow, formula-dense corpora, lexical retrieval is a strong, low-cost default, while hybrid fusion is best reserved for cases where answer relevancy outweighs response time.

Index Terms—BM25, dense retrieval, educational chatbot, hybrid retrieval, quantitative aptitude, query fusion, reranking, Retrieval-Augmented Generation.

I. INTRODUCTION

Conversational AI tutors built on large language models are widely used for self-directed learning, since they can clarify concepts, generate practice

problems, and respond adaptively to a learner's queries at any hour without requiring a human instructor to be present. This flexibility has made LLM-based tutoring systems attractive across school-level and competitive-examination preparation alike. However, an LLM's parametric knowledge alone is an unreliable basis for quantitative subjects: models can fabricate formulas, misstate constants, transpose digits, or silently swap operators, and unlike an error in a descriptive subject, a confidently wrong step-by-step numerical solution can actively reinforce a student's misconception rather than correct it.

Retrieval-Augmented Generation (RAG) addresses this reliability gap by conditioning the generator on passages retrieved from a trusted, inspectable corpus rather than relying purely on the model's internal parameters. Because every claim in the generated answer can, in principle, be traced back to a specific retrieved passage, RAG narrows the space in which a model can hallucinate and gives a tutoring system a natural mechanism for citing its source material. The quality of a RAG system's answers, however, is bounded by the quality of what it retrieves: if the retriever fails to surface the passage containing the relevant formula or worked example, the generator has no correct basis to draw on, however capable it otherwise is.

Prior RAG-based tutors, such as BEEXPERT [1] a voice-based trigonometry tutor pairing dense retrieval with GPT-4o generation report reasonable correctness but only moderate relevancy, attributed partly to retrieval quality and context fragmentation during chunking, while committing to a single (dense) retrieval strategy without testing alternatives. This is a common pattern in educational RAG systems

generally: a retrieval strategy is chosen once, often because it is the default in a given framework, rather than validated against the linguistic structure of the target domain. Quantitative-aptitude content is dense with numerals, symbols, units, and short formulaic phrases that are often better matched by exact lexical overlap than by semantic embedding similarity, raising a practical question this project investigates directly: is dense, sparse (BM25), or hybrid retrieval better suited to this domain, and what trade-offs in accuracy, evidence quality, and latency does each impose?

This project implements one tutoring pipeline with three interchangeable retrieval back-ends BM25, dense, and hybrid sharing an identical corpus, chunking configuration, cross-encoder reranker, and generator LLM, so that any difference in downstream outcome is attributable to the retrieval strategy alone rather than to confounding implementation differences. The pipeline is evaluated on a 1000-question benchmark across sixteen quantitative-aptitude topics using retrieval-quality metrics (precision, recall, context relevancy) and generation-quality metrics (answer relevancy, exact-match accuracy, ROUGE-L), alongside end-to-end latency. The target use case is a self-study companion for learners preparing for school-level and competitive quantitative-aptitude examinations, where questions are typically short, numerically precise, and drawn from a bounded set of formula families rather than open-ended discursive topics. This bounded, formula-centric setting is precisely where the choice between lexical and semantic retrieval is least obvious a priori, which is what motivates treating it as an empirical question rather than assuming one retriever is universally superior.

This project also differs from prior educational RAG deployments in evaluation posture. BEEXPERT [1], for instance, foregrounds an interactive, voice-based experience and validates the resulting system primarily through an LLM-judge correctness/faithfulness/relevancy protocol together with a Godspeed-based feasibility questionnaire capturing likeability, perceived intelligence, and anthropomorphism from a small pool of human evaluators. That design answers a different question whether users find a single, already-built configuration usable and trustworthy rather than which retrieval strategy a designer should choose in the first place.

The present work instead treats retrieval strategy itself as the object of study: three otherwise-identical pipelines are run to completion over a shared, fixed benchmark so that precision, recall, context relevancy, answer relevancy, exact-match accuracy, ROUGE-L, and latency can be compared directly and attributed solely to the retrieval mechanism, before any single configuration is committed to for interactive deployment.

The remainder of this report proceeds as follows. Section II reviews related work in RAG and retrieval-strategy comparison. Section III describes the system architecture. Section IV details the experimental setup. Section V presents and discusses results. Section VI outlines limitations, and Section VII concludes with future directions.

II. RELATED WORK

Retrieval-Augmented Generation was introduced by Lewis et al. [2], who combined a pre-trained sequence-to-sequence generator with a dense passage retriever trained end-to-end, showing that conditioning generation on retrieved evidence improves factual accuracy on knowledge-intensive tasks relative to purely parametric generation. Around the same period, REALM [3] integrated retrieval directly into language model pretraining rather than adding it only at inference time, while Fusion-in-Decoder [4] took a complementary approach of encoding each retrieved passage independently and fusing their representations during decoding an alternative to this project's simpler strategy of concatenating the top-ranked chunks into a single prompt context, chosen here to keep the generator interface identical across BM25, dense, and hybrid retrieval so that only the retrieval stage varies. Dense Passage Retrieval [5] established the dual-encoder bi-encoder architecture underlying most dense retrievers used in RAG systems today, in which a query encoder and a document encoder are trained so that relevant query–passage pairs are close in embedding space.

Classical probabilistic lexical scoring such as BM25 [6] remains a strong, low-latency baseline, particularly on entity- and number-dense corpora where exact term overlap carries more signal than semantic similarity a property directly relevant to a corpus built from worked numerical examples. Reciprocal Rank Fusion [7] and related fusion techniques provide a simple,

training-free way to combine ranked lists from heterogeneous retrievers by aggregating rank-based rather than raw scores, sidestepping the problem that cosine similarities and lexical term-frequency scores are not directly comparable; this class of technique forms the basis of the hybrid configuration evaluated in this project.

Within education specifically, BEEPERT [1] is the most directly relevant prior system: a voice-based RAG tutor for high-school trigonometry that indexes an OpenWebMath-derived corpus with a multilingual dense embedding model and generates responses with GPT-4o, evaluated with an LLM judge for correctness, faithfulness, and relevancy, and with the Godspeed Questionnaire Series for user-facing likeability and perceived intelligence. Concretely, BEEPERT filters a 6.3-million-document OpenWebMath collection down to 515 trigonometry-focused sources, splits them into 4,321 chunks, embeds them with the multilingual BGE-M3 model, and stores the resulting vectors in a Supabase-hosted vector database queried purely by dense cosine similarity a single-strategy design with no lexical or fused alternative. BEEPERT reports strong correctness (72.98%) but only moderate relevancy (47.87%) and markedly weaker faithfulness (31.91%) under this dense-only retrieval design, which its authors partly attribute to context fragmentation introduced during chunking and to the retriever's inability to fall back on exact term matching when a query's wording diverges from the source phrasing. Separately, BEEPERT's Godspeed-based feasibility study reports high likeability (88%) and perceived intelligence (80%) but comparatively lower anthropomorphism (75%), suggesting that users judged the system as helpful and competent even where its retrieval-side faithfulness lagged a dissociation between perceived quality and measured groundedness that motivates evaluating retrieval strategy choices independently of surface-level user satisfaction. This project adopts a similar RAG-for-math-tutoring framing but replaces the single dense-retrieval design with a controlled comparison of BM25, dense, and hybrid retrieval, testing directly whether an alternative retrieval strategy narrows the relevancy gap that BEEPERT identifies as a limitation of its own design.

More broadly, retrieval-strategy ablations in the general RAG literature [8], [9] report mixed findings some benchmarks favor dense retrieval, while entity-

centric or fact-heavy benchmarks favor lexical retrieval or hybrid fusion indicating that the best retrieval strategy is domain-dependent rather than universal. Automated RAG evaluation frameworks such as Ragas [10] provide reference-free metrics like answer relevancy that estimate how directly a generated answer addresses the intent of a question, which this project uses alongside more traditional overlap-based metrics to evaluate the generation stage. This project treats the domain-dependence question as its central empirical focus and answers it specifically for the quantitative-aptitude tutoring domain, a setting not previously benchmarked in this way to the best of our knowledge.

Infrastructure choices in the retrieval layer also matter in practice. Efficient approximate nearest-neighbour search libraries such as FAISS [11] make dense retrieval over tens of thousands of chunks tractable at interactive latency, and orchestration frameworks such as LlamaIndex [12] provide the indexing, chunking, and retriever-composition primitives including BM25 retrievers and fusion retrievers used directly in this project's implementation. Building on this tooling rather than a bespoke retrieval stack keeps the three configurations compared in Section V structurally identical apart from the retrieval algorithm itself, which is a precondition for the controlled comparison this project reports. A secondary benefit of relying on established, widely-used libraries rather than custom retrieval code is reproducibility: because BM25, dense, and fusion retrievers are all standard, documented components of the same framework, another practitioner can reconstruct this project's three configurations from the parameters in Table II alone, without needing access to bespoke scoring or indexing code.

The dense encoders and cross-encoders used throughout this pipeline descend from the Transformer architecture [13] and its bidirectional pretraining variant BERT [14], which established contextual token representations as the basis for both bi-encoder embedding models such as Sentence-BERT [15] the architecture underlying bge-small-en-v1.5 and cross-encoder rerankers such as the MiniLM-based reranker used in Section III-C. Alternative late-interaction dense retrievers such as ColBERT [16] preserve token-level representations rather than collapsing a passage into a single vector, trading a larger index footprint for finer-grained query–passage matching;

this project instead uses single-vector dense retrieval for its smaller memory footprint, consistent with the single-accelerator inference budget assumed in Section IV. Approximate nearest-neighbour search over these dense vectors is made tractable at scale by graph-based indexes such as HNSW [17], which underlies the FAISS index configuration listed in Table II.

III. PROPOSED SYSTEM

Fig. 1 summarizes the query-time pipeline: source PDF material is extracted and cached page-by-page, split into overlapping chunks, embedded and indexed

for both dense and sparse retrieval, retrieved under one of three configurations, reranked with a shared cross-encoder, and passed to the generator LLM together with the user's question, before a final answer-extraction and scoring stage. Every stage after chunking is executed identically across the three retrieval configurations, so that the comparison in Section V isolates the retrieval strategy rather than confounding it with differences in preprocessing, reranking, or generation.

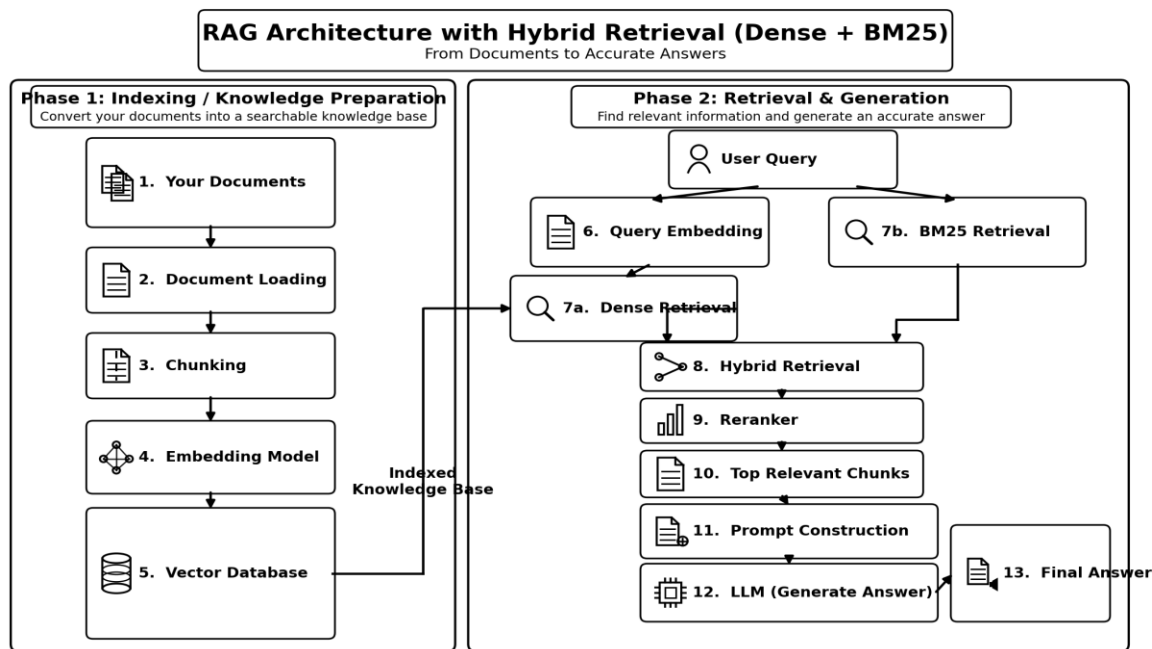


Fig. 1. Query-time pipeline: corpus → chunking → dense/BM25 indexing → retrieval configuration → reranking → generation → scoring.

A. Corpus and Ingestion

The knowledge base is built from a single quantitative-aptitude reference source covering sixteen problem domains commonly tested in competitive and school-level examinations, listed in Table I. Text is extracted page-by-page using PyMuPDF and cached to disk as JSON so ingestion is resumable and need not be repeated across runs. Extracted pages are split into overlapping chunks using a node parser with a chunk size of 900 tokens and an overlap of 120 tokens wide enough to keep a full worked example (problem statement, steps, and final value) largely intact within

one chunk, while the overlap reduces the chance that a formula is cut across a chunk boundary. The accompanying 1000-question evaluation set is drawn from the same reference source rather than authored independently, so that every ground-truth answer is traceable to a specific worked example already present in the indexed corpus; each question-answer pair is checked against the source material before being added to the benchmark, which keeps the evaluation faithful to what the retriever can, in principle, surface.

Table I
Quantitative-Aptitude Topics Covered by the Corpus

Percentage	Number System
Interest (SI/CI)	Mensuration
Ratio & Average	Ages & Alligation
Time & Work	Profit & Loss
Time-Speed-Distance	Permutation & Probability
Data Interpretation	Simplification & Approximation
LCM & HCF	Probability
Boats & Streams	Pipes & Cisterns

B. Dense, Sparse, and Hybrid Retrieval

Chunks are embedded with the bge-small-en-v1.5 sentence-embedding model (384 dimensions) and stored in a FAISS HNSW index ($M = 32$) for approximate nearest-neighbour dense retrieval. Index construction is performed in resumable batches: pages are embedded and inserted a fixed number at a time, with a checkpoint written after each batch, so a large indexing job can be paused and resumed without recomputing embeddings for pages already indexed a practical necessity given the corpus size and shared-runtime constraints of the development environment. In parallel, the same chunk store is indexed for BM25 lexical retrieval, so both retrievers operate over an identical chunk set and only the scoring mechanism differs.

Three retrieval configurations are then compared while holding every other component fixed: (i) Dense-only cosine-similarity search over the FAISS index, returning the top-18 candidates; (ii) BM25-only lexical term-overlap scoring over the same chunk store, also returning the top-18 candidates; (iii) Hybrid a query-fusion retriever that issues the query against both the dense and BM25 retrievers and merges their independently ranked candidate lists into a single fused ranking, so that a chunk ranked highly by either retriever alone has the opportunity to surface in the final candidate set even if the other retriever ranks it poorly. All three configurations retrieve a wide candidate pool before reranking, which favors recall at the retrieval stage and leaves precision to be recovered by the reranker an approach applied identically across configurations for a fair comparison. Fixing the candidate-pool size at 18 for every configuration, rather than letting each retriever return a strategy-specific number of candidates, ensures that any downstream difference in reranked context quality

reflects which chunks each retriever surfaces rather than how many chunks it was given the opportunity to surface, which would otherwise confound the comparison in Section V.

C. Cross-Encoder Reranking

The 18 candidates from each retriever are reranked using a local cross-encoder (cross-encoder/ms-marco-MiniLM-L-6-v2), which scores each query-chunk pair jointly rather than via independent embeddings, typically yielding a more precise relevance ordering than the first-stage retriever alone. The top-3 reranked chunks are kept as the final context for generation. Using a shared, deterministic reranker across all three retrieval configurations ensures that any observed difference in downstream answer quality is attributable to the upstream retriever, not to the reranking stage.

D. Answer Generation

The final context is inserted into a fixed system prompt that instructs the generator to answer strictly from the provided context, to avoid outside knowledge, to render mathematics in plain Unicode notation rather than LaTeX, and to end every response with a single machine-parsable line of the form "Final Answer: <value>". Generation uses a single open-weights LLM (gpt-oss-20b) at temperature 0 to minimize response variance, so the generator is held fixed as a controlled component across all three retrieval configurations. Instruction-tuned decoder-only generators in this lineage from few-shot pretrained language models such as GPT-3 [18], through text-to-text transfer formulations such as T5 [19], to explicit instruction-following fine-tuning such as InstructGPT [20] are trained to follow a system prompt's constraints directly, which is what permits the rigid "Final Answer: <value>" formatting rule in this project's prompt to be enforced reliably rather than merely requested. The system prompt is deliberately restrictive rather than open-ended: forcing a single, bare "Final Answer: <value>" line gives the scoring routine a stable anchor to parse regardless of how much explanatory text precedes it, and banning outside knowledge keeps the generator's output attributable to the retrieved context rather than to the model's own parametric recall of similar problems both choices aimed at making the downstream

comparison in Section V a fair test of retrieval quality rather than of generation style.

E. Answer Extraction and Scoring

Because ground-truth answers in this domain are short a number, a percentage, a ratio automated scoring requires careful normalization rather than raw string equality. The scoring routine extracts the value following “Final Answer:”, strips currency symbols, percentage signs, thousands of separators, and parenthetical notes, and, where the value is numeric or fractional, compares it via exact fraction equality rather than string equality, so that equivalent forms such as “4/5”, “0.8”, and “80%” are correctly scored as matching. This correction was found necessary during development: an early audit of raw results showed that a large share of answers initially marked incorrect were in fact numerically correct but differently formatted from the stored ground truth (for example, an answer prefixed with a currency symbol, missing a trailing percent sign, or annotated with a parenthetical unit). The same audit surfaced a second, subtler failure mode duplicate or restated labels within a single generated answer (e.g., a ratio expressed twice in different but equivalent forms) which the extraction routine resolves by taking the first well-formed value following the “Final Answer:” marker rather than the last, since the generator was observed to occasionally append a redundant restatement after the primary answer.

F. Reproducibility and Interactive Demo

Every evaluation run writes its results to disk after each question, so an interrupted run can resume without repeating already-answered questions a necessary design choice given the scale of the 1000-question benchmark. Independently of the evaluation harness, the same hybrid-retrieval pipeline is exposed as an interactive chat demo, allowing a user to ask a free-form quantitative-aptitude question and receive a live, retrieval-grounded, step-by-step response, which served as a practical check that the pipeline behaves sensibly outside the fixed evaluation benchmark.

IV. EXPERIMENTAL SETUP

A. Implementation Configuration

Table II lists the parameters held fixed across every experiment reported in this paper, so that the ablation

in Section V isolates the effect of the retrieval strategy rather than confounding it with differences in chunking, embedding model, reranker, or generator. Holding these parameters fixed is what makes the three-way comparison in Section V an ablation in the strict sense rather than a comparison of three independently-tuned systems: every reported difference in Table III is attributable to the single varied factor which retriever supplies the reranker's candidate pool rather than to an uncontrolled difference in how each configuration was built.

Table II
System Configuration Parameters

Parameter	Value
Chunking	900 tok / 120 overlap
Dense embedding	bge-small-en-v1.5 (384-d)
Vector index	FAISS HNSW (M=32)
Sparse index	BM25 over docstore
Hybrid fusion	QueryFusionRetriever
Retrieve top-k	18
Reranker	cross-encoder MiniLM-L-6-v2
Rerank top-n	3
Generator	gpt-oss-20b

B. Evaluation Set

The pipeline is evaluated on 1000 question-ground-truth pairs spanning the sixteen topics in Table I. Each configuration (BM25, Dense, Hybrid) is run over the identical question set with retrieval, reranking, generation, and scoring cached to disk after every question, making evaluation runs resumable in the event of interruption, and ensuring that all three configurations are compared on exactly the same set of questions rather than on partially overlapping samples. Embedding, reranking, and generation were executed using the gpt-oss-20b model on a GPU-backed runtime; this keeps every stage of the pipeline local and deterministic, so that reported latency reflects the model's own inference behavior rather than a shared external service.

C. Evaluation Metrics

The following metrics are reported for each configuration.

- Precision / Recall term-overlap-based relevance of retrieved chunks against the ground-truth answer and question, using a fixed overlap threshold to flag a chunk as relevant.

- Context Relevancy the proportion of retrieved chunks judged relevant by the same term-overlap criterion.
- Answer Relevancy a Ragas-computed metric estimating how directly the generated answer addresses the intent of the question.
- Exact Match Accuracy the share of questions for which the normalized final answer exactly matches the normalized ground truth.
- ROUGE-L longest-common-subsequence overlap between the generated answer and the ground truth, capturing partial lexical agreement beyond strict exact match.
- Average Latency mean wall-clock time from query submission to final generated answer, per configuration.

D. Baseline Rationale

BM25 and dense retrieval are the two most widely used single-strategy retrievers in production RAG systems and represent two different, well-understood notions of relevance: lexical overlap and learned semantic similarity making them a natural pair of baselines. Hybrid retrieval is included not as a presumed improvement but as the configuration most commonly recommended in general-purpose RAG guidance as a safe default; testing that assumption directly, rather than adopting it uncritically, is precisely the empirical contribution this project aims to make for the quantitative-aptitude tutoring domain. All three configurations were run to completion on the same evaluation harness under the same generator, so the comparison in Section V reflects genuine differences in retrieved evidence rather than differences in how each configuration was implemented or scored. A fourth, parametric-only configuration generating answers directly from the LLM with no retrieved context at all was deliberately excluded from the main comparison: the project's premise is that some form of retrieval is necessary for a formula-dense tutoring domain, so the scientifically interesting question is which retrieval strategy to use, not whether to use one; a no-retrieval ablation is noted as a natural sanity-check extension in Section VII rather than treated as a fourth peer configuration here.

V. RESULTS AND DISCUSSION

Table III and Fig. 2 report all evaluation metrics for the three retrieval configurations on the 1000-question benchmark. Fig. 3 additionally isolates end-to-end latency, which is the metric with the widest relative spread across configurations.

Table III
Retrieval Configuration Comparison

Metric	BM25	Dense	Hybrid
Avg Latency (s)	0.73	1.54	3.29
Precision (%)	97.67	98.00	96.33
Recall (%)	99.00	100.00	98.00
Ctx. Relevancy (%)	97.67	98.00	96.33
Answer Relevancy (%)	82.50	79.07	87.74
Exact Match Acc. (%)	91.00	86.00	90.00
ROUGE-L (%)	75.96	70.81	73.41

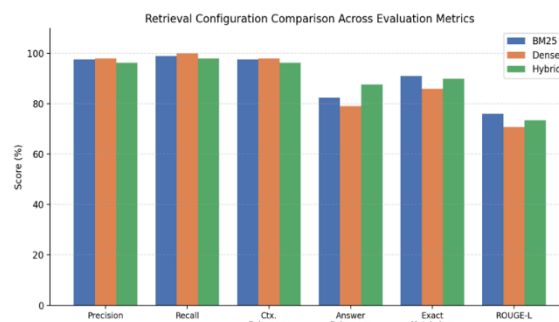


Fig. 2. Retrieval and generation-quality metrics across the three configurations.

A. Retrieval Quality

BM25 and dense retrieval are closely matched on precision, recall, and context relevancy, with dense retrieval marginally ahead on all three (98.00% / 100.00% / 98.00% versus 97.67% / 99.00% / 97.67% for BM25). This is consistent with the corpus characteristics: quantitative-aptitude passages combine short worked formulas, which favor exact lexical matching, with descriptive problem framing, which favors semantic similarity, so neither retriever dominates the other on pure retrieval quality. The hybrid configuration retrieves marginally less precisely and relevantly (96.33% / 98.00% / 96.33%) than either single-strategy retriever, suggesting that fusing two already-strong, largely overlapping candidate lists over this narrow, well-structured corpus does not add retrieval-quality headroom, and can occasionally dilute the top-ranked results relative to

the best single retriever. Qualitatively, the sixteen topics in Table I are not uniformly matched by either retrieval mode: topics such as Percentage, Profit & Loss, and Interest are built from a small, recurring vocabulary of formula terms and constants, where BM25's exact term overlap has an intrinsic advantage; topics such as Ages & Alligation and Time-Speed-Distance instead phrase the same underlying relationship in varied narrative language across different worked examples, a setting where dense retrieval's semantic matching is expected to help close the gap. Table III's aggregate averages necessarily mask this topic-level variation, which is one motivation for the finer-grained, statistically powered topic-level comparison identified as future work in Section VI.

B. Answer Quality

The picture changes at the generation stage. Hybrid retrieval yields the highest answer relevancy (87.74%), ahead of BM25 (82.50%) and dense retrieval (79.07%), despite not leading on raw retrieval-quality metrics. This indicates that the small amount of complementary evidence the hybrid retriever surfaces passages one single-strategy retriever finds but the other does not is disproportionately useful to the generator's judgment of relevance, even where it does not move the strict term-overlap retrieval metrics. Exact-match accuracy tells a partly different story: BM25 attains the highest exact-match accuracy (91.00%) and ROUGE-L (75.96%), with hybrid close behind (90.00% / 73.41%) and dense retrieval trailing on both (86.00% / 70.81%). This gap is consistent with the domain: exact numeric answers depend heavily on the retriever surfacing the specific worked example containing the correct constants and formula, which lexical matching does reliably for this corpus.

C. Latency

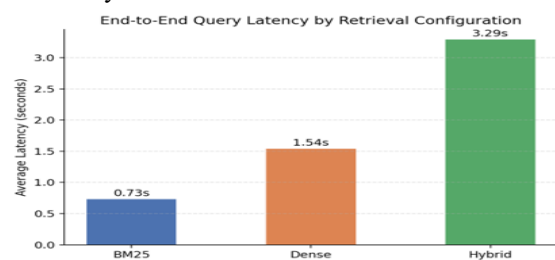


Fig. 3. Mean end-to-end query latency by retrieval configuration.

Latency scales directly with retrieval complexity. BM25 is fastest (0.73 s average), dense retrieval is intermediate (1.54 s), and hybrid is slowest (3.29 s) because it must execute both retrievers and fuse their outputs before reranking and generation proceed. For an interactive tutoring application, a 2.56 s gap between BM25 and hybrid is a meaningful trade-off, since response time directly affects perceived usability during a live tutoring session.

D. Overall Interpretation

No single configuration is uniformly best across every metric. BM25 is the strongest choice when low latency and exact numeric correctness are the priority, which is often the case for a quantitative-aptitude tutor answering short, formula-bound questions. Hybrid retrieval is the strongest choice when answer relevancy how well the response addresses what was actually asked, rather than strict numeric agreement is the priority, at roughly 4.5× the latency of BM25. Dense retrieval, while competitive on raw retrieval-quality metrics, does not lead on any downstream generation-quality metric in this evaluation, suggesting it is the weakest standalone choice for this specific, formula-dense domain, though it remains a necessary component of the hybrid configuration's improved answer relevancy.

E. Comparison with Prior Work

These results provide a partial answer to the gap identified in Section II. BEEPERT [1], using dense-only retrieval on a trigonometry corpus, reports a relevancy score of 47.87% alongside 72.98% correctness. While the two systems differ in domain, corpus, and judge methodology and so are not directly comparable in absolute terms, the qualitative pattern observed here that dense retrieval alone underperforms a fused or lexical alternative on relevancy-style metrics is consistent with BEEPERT's own diagnosis that retrieval quality, rather than the generator, is the primary bottleneck on relevancy in RAG-based math tutors. This project's finding that hybrid fusion measurably improves answer relevancy over dense-only retrieval (87.74% versus a dense-only 79.07% in this setting) offers one concrete, testable direction for narrowing that bottleneck in future iterations of dense-only tutoring systems. It is also notable that BEEPERT's own reported faithfulness score (31.91%) is markedly lower than its correctness

score (72.98%), the same qualitative pattern answers that look right without being verifiably grounded that motivates the faithfulness/hallucination direction identified as future work in Section VII rather than left unaddressed indefinitely. Where BEEPERT additionally reports high user-facing feasibility scores (88% likeability, 80% perceived intelligence) despite this faithfulness gap, the comparison underscores that subjective usability and objective retrieval grounding are separate axes of evaluation that a deployment-ready tutoring system should report side by side rather than substitute one for the other.

F. Practical Deployment Considerations

From a deployment standpoint, the choice among the three configurations is ultimately a product decision as much as a technical one. A self-study application used asynchronously, where a student reviews responses at their own pace, can absorb hybrid retrieval's higher latency in exchange for its relevancy gain. A live, voice-based tutoring session the interaction style demonstrated by BEEPERT is far more latency-sensitive, where a multi-second delay before the first spoken word breaks conversational flow; in that setting, BM25's sub-second latency and strong exact-match accuracy make it the more defensible default, with hybrid retrieval reserved for a fallback path on questions where an initial BM25-only answer scores poorly on an internal confidence check. This kind of tiered retrieval strategy cheap by default, expensive on demand is a natural extension suggested directly by the latency/quality trade-off observed in Section V-C and is noted as a concrete direction in Section VII.

VI. LIMITATIONS

This study has several limitations that bound how far its conclusions should be generalized. First, the evaluation uses a single dense embedding model (bge-small-en-v1.5) and a single generator model family (gpt-oss-20b); results may not generalize to larger or differently trained embedding and generator models, particularly since larger embedding models may narrow or reverse the small margins observed between BM25 and dense retrieval in Section V-A. Second, the 1000-question benchmark, while spanning sixteen topics, still limits the statistical power of any fine-grained topic-level comparison that a still larger study might otherwise support.

Third, latency was measured for a single, locally-served deployment of gpt-oss-20b, which may not reflect latency under a different serving configuration or hardware, and could over- or under-state the true latency gap between configurations in a production setting. Finally, the hybrid configuration uses a single fusion method (query-fusion retrieval); alternative fusion schemes such as reciprocal rank fusion with tuned weighting were not compared and may narrow the latency-quality trade-off observed here. A related limitation concerns the evaluation's exclusively automated character: unlike BEEPERT's Godspeed-based feasibility study, this project reports no human-in-the-loop usability judgments (likeability, perceived intelligence, or ease of use), so it is not yet known whether the retrieval strategy that performs best on the automated metrics in Table III is also the one learners would judge as most helpful in practice a gap that a small human evaluation, similar in spirit to BEEPERT's, could close in future work.

A further limitation concerns statistical rigor: the metric differences reported in Table III and Section V are descriptive averages over the 1000-question benchmark rather than the output of a paired significance test, so small gaps such as the 1–2 point differences in precision and recall between BM25 and dense retrieval should be read as indicative rather than conclusive. A larger benchmark with paired statistical testing across configurations, similar in spirit to the significance analysis used in broader retrieval-ablation studies, would be needed to state these differences with confidence.

VII. CONCLUSION AND FUTURE WORK

This project implemented and compared BM25, dense, and hybrid retrieval within an otherwise identical RAG pipeline for a quantitative-aptitude tutoring chatbot, holding corpus, chunking, reranking, and generation fixed across all three configurations. On a 1000-question benchmark spanning sixteen topics, BM25 and dense retrieval achieved comparable, marginally superior retrieval-quality metrics, while hybrid retrieval produced the highest answer relevancy at the cost of substantially higher latency, and BM25 delivered the best exact-match accuracy and lexical overlap with ground truth at the lowest latency of the three configurations. These results support choosing BM25 as a strong, low-cost

default for narrowly-scoped, formula-dense tutoring domains, and reserving hybrid retrieval for scenarios where answer relevancy is prioritized over response time.

Future work includes: evaluating additional embedding models and fusion strategies such as reciprocal rank fusion with tuned weights; expanding the benchmark with paired statistical testing to support finer-grained topic-level comparison; running a small human feasibility study covering likeability, perceived usefulness, and ease of use in the manner of BEEPERT's Godspeed-based evaluation to check whether the automated-metric winner in Table III is also the configuration learners prefer in practice; adding a no-retrieval, parametric-only baseline as a sanity check on how much of the observed answer quality is attributable to retrieval at all; incorporating a faithfulness/hallucination metric so that a correct-looking answer can be confirmed as genuinely grounded in retrieved context rather than merely well-formatted; and extending the pipeline with voice-based interaction and a digital canvas for handwritten input, following the interaction design demonstrated in comparable RAG-based tutoring systems such as BEEPERT [1].

More generally, this project's central takeaway for practitioners building similar domain-specific tutoring systems is that retrieval-strategy choice should not be inherited by default from a framework or from results reported on a different domain; the comparison in Section V shows that even within a single narrow subject area, the best retriever for retrieval-quality metrics (dense) is not the best retriever for downstream answer quality (BM25 or hybrid, depending on which quality dimension is prioritized), underscoring the value of a small, targeted ablation like this one before committing to a production retrieval design.

ACKNOWLEDGMENT

I express my sincere gratitude to Prof. D. Lalitha Bhaskari for her invaluable guidance, constant encouragement, and support throughout the completion of this project.

REFERENCES

- [1] M. Marchella *et al.*, "BEEPERT: Voice-based chatbot with RAG for personal high school mathematics tutor," in *Proc. ICoICT*, 2025.
- [2] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. NeurIPS*, 2020.
- [3] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "REALM: Retrieval-augmented language model pre-training," in *Proc. ICML*, 2020.
- [4] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," in *Proc. EACL*, 2021.
- [5] V. Karpukhin *et al.*, "Dense passage retrieval for open-domain question answering," in *Proc. EMNLP*, 2020.
- [6] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends Inf. Retrieval*, vol. 3, no. 4, 2009.
- [7] G. V. Cormack, C. L. A. Clarke, and S. Buettcher, "Reciprocal rank fusion outperforms Condorcet and individual rank learning methods," in *Proc. ACM SIGIR*, 2009.
- [8] Y. Gao *et al.*, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, 2023.
- [9] Z. Rackauckas, "RAG-Fusion: A new take on retrieval-augmented generation," *arXiv preprint arXiv:2402.03367*, 2024.
- [10] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "RAGAs: Automated evaluation of retrieval-augmented generation," in *Proc. EACL*, 2024.
- [11] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *arXiv preprint arXiv:1702.08734*, 2017.
- [12] J. Liu, "LlamaIndex: A data framework for your LLM applications," 2022. [Online]. Available: GitHub repository
- [13] Vaswani *et al.*, "Attention is all you need," in *Proc. NeurIPS*, 2017.
- [14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep

- bidirectional transformers for language understanding,” in *Proc. NAACL-HLT*, 2019.
- [15] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using Siamese BERT-networks,” in *Proc. EMNLP-IJCNLP*, 2019.
 - [16] O. Khattab and M. Zaharia, “ColBERT: Efficient and effective passage search via contextualized late interaction over BERT,” in *Proc. ACM SIGIR*, 2020.
 - [17] Y. A. Malkov and D. A. Yashunin, “Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 824–836, 2020.
 - [18] T. Brown *et al.*, “Language models are few-shot learners,” in *Proc. NeurIPS*, 2020.
 - [19] Raffel *et al.*, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *J. Mach. Learn. Res.*, vol. 21, no. 140, 2020.
 - [20] L. Ouyang *et al.*, “Training language models to follow instructions with human feedback,” in *Proc. NeurIPS*, 2022.