TechDoc-RAG: An Evidence-Governed Framework for Software Technical Documentation

Sejal Rode[1], Anuradha Shinde[2]
[1,2]*School of Information Technology, Indira University*
doi.org/10.64643/ijirt.208447-459

***Abstract*—Technical support repositories contain large numbers of manuals, technotes, and troubleshooting records, yet their value depends on whether a user can locate and trust the evidence behind an answer. This study develops TechDoc-RAG, an evidence-governed retrieval-augmented generation pipeline for software documentation. Its retrieval layer joins dense semantic search with BM25 lexical search by Reciprocal Rank Fusion (RRF); its response layer separates answerability assessment, corrective retrieval, grounded generation, citation validation, and post-generation support checks. The experiments use NVIDIA TechQA-RAG-Eval and the related TechQA technical-document corpus. On a fixed 100-question retrieval test, Hybrid RRF raised Recall@1 from 0.39 to 0.43, Recall@3 from 0.50 to 0.60, Recall@5 from 0.60 to 0.64, and MRR@5 from 0.4608 to 0.5178 when compared with dense retrieval. The tested cross-encoder reranker added substantial delay without improving the main ranking outcomes. On the separately locked 50-question generation test, the final pipeline increased citation validity from 23.33% to 60% and final refusal accuracy from 0% to 55%, reduced average latency from 1231.26ms to 407.53ms, and eliminated empty outputs. Unsupported outputs decreased from 28 to zero according to the implemented validation criteria. Answer semantic similarity fell from 0.2741 to 0.2356 and the context-support similarity proxy fell from 0.4723 to 0.4111. These findings expose a practical trade-off: stricter evidence controls improve traceability and abstention behavior, but do not improve every automatic semantic measure.**

***Index Terms*—Retrieval-Augmented Generation, Software Documentation, Hybrid Retrieval, BM25, Reciprocal Rank Fusion, Evidence Validation, Answerability Detection**

I. INTRODUCTION

Technical support artifacts are created by modern software products on a large scale and in many different forms, such as per-version technical documentation, error diagnosis, API documentation, product manuals and installation instructions. However, this information doesn't readily lend itself to being used as a 'Knowledge Base' and can answer many user questions. A lot of queries are expressed in natural language, and they are answered by, for example, a command line option, a release version number, or an error message code. The retrieval problem is not just a problem of matching of tokens. Retrieval-augmented generation (RAG) is an appealing alternative that supplies a generation model with documents connected to a provided query, instead of expecting that it can respond to it simply cross its learned parameters [1]. But technical support requires retrieved context to be meaningful and understandable. The answer that is created might appear to follow a pattern and have fewer relevant contexts. Furthermore, even if the technical documentation does not contain an exact or "partial" match, weak or partial matches can yield plausible answers, even though they are factually incorrect or not responsive to a specific product operation described in a question. For technical documents, there are two main signals that are important to have to be effective at RAG. In cases where the wording differs, semantic embedding retrieval can map between them in a question and the relevant documentation. Lexical search (such as BM25) is helpful when a question is referring to a specific configuration parameter, option, product entity or error message. In addition to retrieval, a good system should include processes of assessing the reliability of the evidence, connecting evidence to original sources and re-processing and/or rejecting evidence material that is not valid based on some criteria. The proposed system, TechDoc-RAG, is a staged workflow: It integrates information obtained from the dense embeddings and BM25, and decides whether the answer is answerable based on the expanded set of relevant snippets, allows for one

guided correction step, restricts the generation of documents to only involve the documents retrieved, verifies the citations, checks whether there is some other piece of supporting evidence for every factual statement, and either returns a corrected and evidence-backed document generated answer or a verified and automatically extracted fallback answer or refusal.

We make 4 contributions principally:

1. A new, working RAG pipeline for technical documentation for both dense (semantic) search and token matching.
2. The second controlled study is done between retrieval approaches for the system (dense, RRF, RRF + Cross-Encoder re-ranking).
3. An answer generation pathway that is evidence-based. This way, evidence retrieval is divorced from answerability judgment, citation verification and answer-body evidence support.
4. An evaluation that is frozen and that provides a concise summary of performance as well as measures that provide information about trade-offs to reliability.

II. RELATED WORK

2.1 Retrieval-Augmented Question Answering

The key idea of RAG is to provide information from some external database to help the model [1]. It might be helpful in some situations in which the corpus is very large and continually updated and which might call for a citation of specific knowledge. TechQA established an evaluation setting rooted in technical-support question answering [3]; NVIDIA TechQA-RAG-Eval adapts this material for RAG-oriented evaluation [10]. The present work uses that setting to study not only retrieval success but also the decision to answer or abstain.

2.2 Complementary Retrieval Signals

Dense Passage Retrieval maps queries and passages into a shared representation space, allowing lexical differences to be treated as related when their meanings are close [2]. Sentence-BERT makes sentence-level embedding comparison practical [8], and FAISS provides efficient similarity search over vector collections [9]. In contrast, BM25 scores the terms present in documents and queries using a probabilistic ranking framework [11]. Neither signal fully replaces the other for software documents: embeddings help with paraphrase, whereas term matching preserves evidence from exact technical vocabulary. RRF is used here to combine the two ranked lists without forcing their scores onto a common scale.

2.3 Retrieval Correction and Response Evaluation

A second stage ranker can re-rank the pre-selected candidate list, and a typical example is RankRAG, which is a RAG research based on ranking [7]. Empirically, one would have to determine whether the additional computation leads to an improvement in performance of the corpus. But, in addition to the semantic question and answer comparison, deeper research such as RAGAS [5] implies that an answer may not be similar to the reference answer, and Corrective RAG [6] suggests that before generating an answer, it is worthwhile to explore the relevance of retrieved documents and correct the passage generation. These developments inform TechDoc-RAG's evidence adequacy analysis, citation validation and controlled abstention.

2.4 Motivation for the Present Study

There are some building blocks in the literature, but we'd like to have some building blocks for technical support assistants, which are interactive. We are currently investigating the interaction between the hybrid retrieval and evidence controls – does lexical-semantic fusion improve the ability to find the candidate passage and how do source and answerability checks affect the behavior of the answer generator?

III. METHODOLOGY AND PROPOSED SYSTEM ARCHITECTURE

We see a technical answer as a series of decisions based on evidence not just a single LLM completion at the TechDoc-RAG level. First, we use the retriever to get the set of candidate passage; next a gate determines if these passages can be sufficient; only the questions deemed sufficient enter the generation stage; and all the generated responses are checked against both their source and their support. The final operational architecture we come up with is shown in Figure 1.

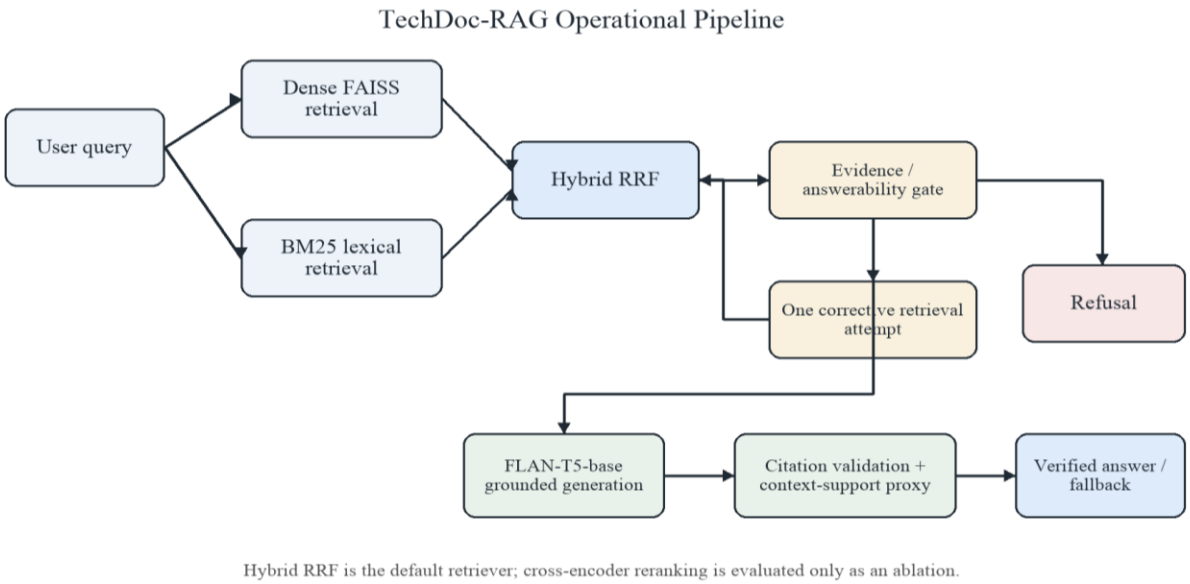

Figure 1. Evidence guided TechDoc-RAG workflow, where the hybrid RRF is used as our deployed retriever and the scorer is only applied for ablations.

3.1 Corpus Preparation and First-Stage Retrieval

Technical document corpus is normalised into passages of length ~180 words and overlap of length 30 words. Persistent document and chunk identifier in each passage. It's passed down to subsequent search result lookups and subsequent cross-linking to the search results. All-MiniLM-L6-v2 will encode queries and chunks for dense retrieval, before FAISS IndexFlatIP will retrieve the top highly similar candidates. [8], [9]. The lexical BM25 retrieval is performed by bm25s on tokenized documentation and the tokenization of certain terms (e.g., error strings, Command names) is intentionally not removed because it is likely to be important for technical support.

These are different as they return orderings (rankings) instead of directly comparable numeric scores. For a document d, TechDoc-RAG computes $RRF(d) = \Sigma_r 1/(k + rank_r(d))$, with k set to 60. The configuration considers 100 candidates and selects the top five final passages. This approach promotes passages that rank well in either or both retrieval systems. The locked test selected Hybrid RRF as the operational retriever because it gave the best balance of early-rank retrieval quality and latency.

3.2 Reranking as an Ablation

The study additionally applies cross-encoder/ms-marco-MiniLM-L6-v2 to the top 20 Hybrid RRF candidates. A cross-encoder jointly reads a query and a candidate passage, so it is more computationally expensive than independent first-stage scoring. The component is kept outside the deployed path. Its purpose is to test whether additional relevance modeling improves this corpus, not to assume that a reranker must be beneficial.

3.3 Evidence Gate and Corrective Search

The answerability mechanism has two frozen stages. Stage A is a standardized, class-balanced logistic-regression classifier using features derived from dense similarity, RRF behavior, lexical retrieval, query-to-context similarity, and agreement between retrieval methods. Stage B is an evidence-quality assessment using RRF score structure, top-five source overlap between dense and BM25 retrieval, semantic query-context similarity, and technical lexical overlap. The frozen decision thresholds are 0.345941 for Stage A and 0.498696 for Stage B. Both stages must pass before the system proceeds. When either stage fails, the pipeline performs one deterministic corrective Hybrid RRF pass and recomputes the decision features. If the evidence submitted is not adequate, the message “Insufficient information in the provided documentation.” is returned. Search can be called at most at most, and can be tried at most at most, avoiding predictable effects from repeated search calls to bound the total time of search.

3.4 Generation, Citations, and Support Checks

Google/flan-t5-base is primed with a well supported Question to generate a grounded, short answer, using the supported snippets included.

The generator's knowledge is limited to the documentation supplied. When the evidence exists but there is no evidence identifier, the identifier is not included in the references provided as usual, but can be added during the process of comparing the evidence generated by the model and the evidence supported by the identifier. The tool can calculate the similarity of the sentence generated after the generation step in terms of context support similarity, which is a good indicator but cannot ensure the faithfulness of content in the generated sentence.

A failed citation or support check triggers at most one stricter regeneration; a remaining failure becomes a verified extractive response or a refusal. Consequently, the term unsupported output in this paper is limited to the implemented evidence and source-validation criteria.

IV. EXPERIMENTAL SETUP AND EVALUATION METRICS

4.1 Dataset, Splits, and Reproducibility Controls

NVIDIA TechQA-RAG-Eval and its associated TechQA technical-document corpus provide the experimental material [10]. Approximately 28,481 documents become approximately 99,216 chunks after processing. Retrieval performance is measured on a fixed 100-question sample. Generation is measured once on an independent locked set of 50 questions: 30 labelled answerable and 20 labelled unanswerable. Sampling and calibration use seed 42. The selected test identifiers, baseline outputs, and final configuration were preserved without result-driven tuning after the final run.

A separate 280-question development pool calibrates the answerability controls and does not overlap with either locked test set. The frozen system consists of dense FAISS retrieval, bm25s Lucene BM25, Hybrid RRF with k = 60, candidate depth 100, and top-k 5; the two evidence thresholds from Section 3; google/flan-t5-base on CPU; and a context-support proxy threshold of 0.25. The cross-encoder reranker is deliberately absent from the default route.

4.2 Retrieval Measures

Recall@1, Recall@3, and Recall@5 indicate whether a relevant passage appears within the first one, three, or five retrieved items. MRR@5 records the reciprocal position of the first relevant result within the top five and contributes zero when no relevant result appears there. Mean retrieval latency is reported alongside these ranking measures because a slower system may not be a practical improvement.

4.3 Generation and Reliability Measures

The generation comparison records answer semantic similarity, context-support similarity proxy, citation validity, final refusal accuracy, average latency, empty-output count, and unsupported-output count. Answer similarity is the cosine similarity of embeddings of generated and reference answers, which assesses the semantic similarity but not the final correctness of the answer. Similarly, context-support measure is a proxy and not called faithfulness.

Citation validity is a score on responses that can be successfully located in evidence-support documentation source.

Refusal accuracy measures the percentage of questions that were not answered by the child that are not answered in the test data, that is, percentage of unanswered questions in the test data that are also not answered in the refusal data. The evidence rule or the source-validation rule will be used to determine whether an answer is unsupported.

An answer will be considered to be unsupported if it does not meet either the evidence rules or the source-validation rule. Zero unsupported indicates no output that was returned that was not supported. It does not establish universal factual correctness or complete claim-level grounding.

4.4 Integrity of the Final Evaluation

Calibration and locked evaluation were separated. Thresholds were fixed before final generation evaluation, all 50 locked questions were executed once, poor cases remained in the results, and the reranker ablation was retained despite its unfavorable outcome. These controls preserve the distinction between development decisions and reported evaluation results.

V. RESULTS AND ANALYSIS

Table 1. Retrieval performance on the locked 100-question evaluation set; higher values are preferable except for latency.

Approach	R@1	R@3	R@5	MRR@5	Latency (ms)
Dense	0.39	0.50	0.60	0.4608	19.88
Hybrid RRF	0.43	0.60	0.64	0.5178	23.20
Hybrid RRF + reranker	0.40	0.53	0.64	0.4883	907.41

Note. *The selected operational retriever is Hybrid RRF. The reranking configuration is reported only as an ablation*

Table 2. Generation behavior on the locked 50-question evaluation: baseline and frozen TechDoc-RAG v4.

System	Answer similarity	Context-support proxy	Citation validity	Refusal accuracy	Avg latency (ms)	Empty	Unsupported
Baseline FLAN-T5-small	0.2741	0.4723	23.33%	0%	1231.26	4	28
Frozen TechDoc-RAG v4	0.2356	0.4111	60.00%	55%	407.53	0	0

Note. *Context-support is an embedding-based similarity proxy, not a complete faithfulness score. The unsupported-output count is defined by the implemented evidence and source-validation checks.*

Hybrid RRF is the strongest first-stage retriever in Table 1. Against dense retrieval, its Recall@1 rises by 0.04, Recall@3 by 0.10, Recall@5 by 0.04, and MRR@5 by 0.0570. The additional mean latency is only 3.32 ms. In a corpus where technical tokens and natural-language descriptions both matters, this result supports combining lexical and semantic rank evidence.

The reranking ablation changes the cost profile dramatically without producing a ranking gain. Recall@3 declines from 0.60 to 0.53 and MRR@5 declines from 0.5178 to 0.4883 relative to Hybrid RRF. Retrieval latency reaches 907.41ms instead of 23.20ms. Although Recall@5 remains 0.64, the cost and lower earlier-rank performance justify retaining the reranker only as an experimental comparison.

Table 2 shows a different kind of trade-off. Frozen TechDoc-RAG v4 improves citation validity from 23.33% to 60% and changes final refusal accuracy from 0% to 55%. Its average latency falls from 1231.26ms to 407.53ms; empty outputs fall from four to zero; and unsupported outputs decrease from 28 to zero according to the implemented validation criteria. Those are just the changes in behaviour of evidence management and evidence traceability; it isn't evidence that everything is made more accurate.

In contrast, automatic semantic scores decrease, ranging from 0.2741 to 0.2356 (see changes in answer similarity) and from 0.4723 to 0.4111 (see changes in context-support similarity proxy). Words generated through all three – conservative abstention, citation management and extractive fallback – can of course differ in wording from the words in the reference answer. Moreover, the context-support proxy should not be read as faithfulness. The final gate directly refused 9 of 20 unanswerable questions; after corrective retrieval and the complete pipeline, 11 of 20 were finally refused. Therefore 55% is a marked improvement over the baseline but remains a clear limitation. The run produced 25 model-generated factual answers, two extractive fallbacks, 23 final refusals, six strict regenerations, and no technical failures.

VI. DISCUSSION

6.1 Interpreting the Retrieval Outcome

Technical queries often mix two forms of signal. A user may paraphrase the problem, which favours dense representations, while also naming a precise parameter or product feature, which favours BM25. RRF gives both signals a chance to contribute through their rank positions. The improvement at Recall@3 and

MRR@5 indicates that the fused list more often makes useful evidence available near the front of the generator's context window.

6.2 Interpreting the Reranker Outcome

The evaluated reranker did not earn its computational cost in this experiment. This result does not establish that reranking fails in all RAG systems. It identifies a mismatch between the chosen cross-encoder, the Hybrid RRF candidate set, and this technical-support corpus. Reporting the negative ablation is important because it supports a simpler final path based on measured behavior rather than assumed complexity.

6.3 Reliability Versus Semantic Resemblance

The final pipeline adds explicit decisions that an instruction-only baseline does not make: whether there is enough evidence to answer, whether a source identifier exists in the retrieved set, and whether a response clears the implemented support rule. These increases in citations, final refusals, empty output prevention and validation-defined unsupported outputs are beneficial for documentation support. This decrease in the similarity of answers further demonstrates that there is a design trade-off between controlled reliability. TechDoc-RAG should thus be interpreted as a system that aims to instill evidence-aware behaviour, rather than one that prioritizes all the generation metrics.

VII. LIMITATIONS AND FUTURE WORK

This assessment is restricted to certain scales and available automatic measures. The locked generation set contains a total of 50 questions, and allows an unbiased comparison of the controlled situation, but cannot describe all technical-documentation domains. The last 50% accuracy of refusal is flawed: it sometimes answers a question that has no answer, and it sometimes refuses an answerable question. Both answer semantic similarity and the context-support similarity proxy are indirect signals, the latter does not check each individual claim against source text.

The generator, FLAN-T5-base on CPU, is a controlled and reproducible choice rather than a state-of-the-art instruction-following system. Future work should repeat the same controls with stronger models, larger held-out tests, more diverse documentation, and human or claim-level grounding assessment. Better answerability features and calibration may improve the answer/refusal boundary. Version-aware retrieval is another important direction for documents that vary by software release, but it was not evaluated in the present experiment and is therefore not claimed as a result.

VIII. CONCLUSION

TechDoc-RAG combines hybrid retrieval with explicit evidence governance for software technical documentation. Dense FAISS search and BM25 lexical search are fused by RRF; answerability is assessed before generation; and citations plus a context-support proxy are checked before a response is released. This design makes the system's evidence decisions visible rather than leaving all reliability decisions to a generator.

The fixed evaluation selected Hybrid RRF as the final retriever. It improved retrieval metrics over dense search while remaining low latency, whereas the tested reranker was slower and did not improve the main ranking measures. The frozen generation pipeline increased citation validity and final refusal accuracy, eliminated empty outputs, and reduced unsupported outputs from 28 to zero according to the implemented validation criteria. It also reduced answer similarity and the context-support similarity proxy, and its 55% refusal accuracy remains limited. These results support an evidence-aware approach to technical-document answering while keeping its trade-offs explicit.

REFERENCES

[1] P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[2] V. Karpukhin et al., “Dense passage retrieval for open-domain question answering,” in Proc. 2020 Conf. Empirical Methods in Natural Language Processing, pp. 6769–6781, 2020.

[3] V. Castelli et al., “The TechQA dataset,” in Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, pp. 1269–1278, 2020, doi: 10.18653/v1/2020.acl-main.117.

[4] S. Zhou, U. Alon, F. F. Xu, Z. Wang, Z. Jiang, and G. Neubig, “DocPrompting: Generating code by

retrieving the docs," in Int. Conf. Learning Representations, 2023.

[5] S. Es, J. James, L. Espinosa Anke, and S. Schockaert, "RAGAS: Automated evaluation of retrieval augmented generation," in Proc. 18th Conf. Eur. Chapter Assoc. Comput. Linguistics: System Demonstrations, pp. 150–158, 2024, doi: 10.18653/v1/2024.eacl-demo.16.

[6] S. Q. Yan, J. C. Gu, Y. Zhu, and Z. H. Ling, "Corrective retrieval augmented generation," arXiv preprint arXiv:2401.15884, 2024.

[7] Y. Yu et al., "RankRAG: Unifying context ranking with retrieval-augmented generation in LLMs," Advances in Neural Information Processing Systems, vol. 37, 2024.

[8] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proc. 2019 Conf. Empirical Methods in Natural Language Processing, pp. 3982–3992, 2019.

[9] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," IEEE Trans. Big Data, vol. 7, no. 3, pp. 535–547, 2021.

[10] NVIDIA Corporation, "TechQA-RAG-Eval," Hugging Face dataset, May 2025. [Online]. Available: . Accessed: Aug. 24, 2026.

[11] 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.