

RAG-MedDx: A Preliminary Offline Prototype of a Citation-Traced Retrieval-Augmented Framework for Intelligent Medical Diagnosis Support

AKANKSHA Gaikwad

`akanksha.gaikwad_24uds@sanjivani.edu.in`

sanjivani university

Research Article

Keywords: Retrieval-Augmented Generation, Large Language Models, Intelligent Medical Diagnosis, Citation Traceability, Offline Prototype, Clinical Decision Support Systems

Posted Date: October 28th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11514301/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

Abstract

Large Language Models (LLMs) can generate fluent medical text, but a standalone LLM relies on parametric knowledge alone, so it may produce incorrect diagnostic statements that cannot be traced to a verifiable source. Retrieval-Augmented Generation (RAG) grounds output in external knowledge, yet existing medical RAG studies address retrieval technique, accuracy, or explainability in isolation. This paper presents a preliminary offline prototype, RAG-MedDx, of a four-layer RAG framework (retrieval, augmentation, generation, and traceability) for medical diagnosis support. The prototype retrieves from a local knowledge base of six textbook-tagged entries using keyword matching, composes rule-based answers with inline citations to the retrieved sources, and reports that no source was found, instead of generating an answer, when a query matches nothing. On two sample queries it returned a citation-traced answer in 0.77 s (2 sources retrieved, 4 query terms matched) and a no-match message in 0.49 s (0 sources retrieved). A Python framework that compares a standalone LLM with a RAG-LLM pipeline on accuracy, hallucination rate, citation coverage, and latency was also implemented; it passed an offline functional test with a mock model but has not yet been run with a real LLM. The evaluation is preliminary: the prototype uses a simplified generator instead of an LLM, accuracy, hallucination rate, and explainability were not measured, and the system is not a clinical tool. The work demonstrates the feasibility of citation-traced, no-match-aware retrieval only and makes no claim of improved diagnostic performance.

I. INTRODUCTION

Artificial intelligence (AI) has become a powerful tool across many industries, including healthcare, education, and banking. Current AI systems rely heavily on large language models (LLMs) for reasoning, natural-language understanding, and decision support. Despite their fluency, these models generate text by next-token prediction, which can yield content that has not been fact-checked. This property is especially dangerous when LLMs are applied to medical diagnosis.

Two issues dominate discussions of LLMs in high-stakes domains. The first is hallucination: the model produces fluent, confident output that is factually incorrect or unsupported by medical evidence. A hallucinated drug interaction or an incorrect symptom–disease mapping in a diagnostic suggestion becomes a patient-safety problem. The second is opacity. LLMs largely behave as black boxes, and they offer no means of tracing the reasoning behind a generated output.

RAG addresses both issues directly by grounding generation in retrieved documents instead of relying solely on parametric memory. A review of related work, however, shows that existing medical RAG applications typically tackle only one problem at a time, whether retrieval method, diagnostic accuracy, or reduced misdiagnosis. They do not build an integrated system that evaluates accuracy, reliability, and explainability together against an LLM baseline.

This research therefore presents a preliminary prototype of an Intelligent Medical Diagnosis System based on RAG and LLMs. Whereas prior work concentrates on individual modules in isolation, the

present work designs an integrated RAG–LLM diagnostic pipeline, implements a simplified offline prototype of it, and builds an evaluation framework for a later comparison with a standalone LLM. The framework aims to reduce hallucinations and improve the explainability of LLM-generated medical diagnoses; these effects are not yet measured in this paper. The research objectives are:

- To design and develop a RAG framework that grounds the LLM’s diagnostic suggestions in verified medical sources;
- To implement a traceable and explainable diagnostic pipeline in which every suggestion links back to its supporting evidence; and
- To evaluate the accuracy, reliability, and efficiency of the proposed RAG–LLM system against a standalone (non-RAG) LLM baseline.

The remainder of this paper is organized as follows. Section II reviews the literature, Section III states the research gap and objectives, Section IV describes the proposed framework and planned evaluation, Section V describes the system implementation, Section VI presents the results of the prototype demonstration, a discussion, and the planned quantitative evaluation, Section VII discusses limitations, and Section VIII concludes the paper.

II. RELATED WORK

This section reviews ten recent studies on Retrieval-Augmented Generation for medical diagnosis and clinical decision support. Each study is summarized together with the limitation that is relevant to the present work. Table 1 gives a condensed comparison.

A. RAG for Differential Diagnosis and Biomedical Surveys

Pinna et al. [1] built an LLM-based differential diagnosis assistant that uses RAG with symptom-focused context extracted from PubMed abstracts. Evaluators rated it more trustworthy than plain ChatGPT-4 because it cited its sources. The study, however, concentrates on comparing the clarity of exposition between models. It does not systematically benchmark hallucination reduction, diagnostic accuracy against a standalone baseline, or response latency, so the trade-off among accuracy, reliability, and efficiency remains unquantified.

He et al. [2] surveyed the biomedical RAG landscape, classifying systems into naive, advanced, and modular paradigms. They also formalized a “biomedical RAG trilemma” among reasoning depth, inference latency, and data privacy. Because the paper is a survey, it neither implements nor experimentally validates a specific diagnostic pipeline, and practical solutions that balance the trilemma remain unexplored.

B. Retrieval Techniques and Knowledge Structures

Liang et al. [3] proposed RGAR, which retrieves both factual knowledge from electronic health records and conceptual medical knowledge, allowing the two sources to refine each other. Notably, it enables a smaller open model (Llama-3.1-8B-Instruct) to outperform a larger RAG-enhanced GPT-3.5. The work centers on retrieval technique and factual accuracy. It does not address explainability or source traceability for clinician-facing use, and it does not evaluate latency overhead.

Zhao et al. [4] presented MedRAG, which constructs a four-tier hierarchical diagnostic knowledge graph so that the RAG model can distinguish between diseases with similar manifestations, with the direct aim of reducing misdiagnosis. Although diagnostic specificity improves, the paper does not benchmark the system against a standalone (non-RAG) LLM across accuracy, reliability, and efficiency together, and explainability is not assessed through structured clinician feedback.

Ansari et al. [5] combined a small, QLoRA-fine-tuned Llama 3.2-3B-Instruct model with RAG grounded in hospital-specific data. Their results show that a lightweight, quantized model can perform reasonably on medical benchmarks while remaining computationally accessible for low-resource hospital settings. The study is oriented toward deployment efficiency and does not focus on tracing the diagnostic reasoning chain back to its source evidence.

C. Clinical Evaluation and Benchmarking

Zakka et al. [6] introduced Almanac, which augments an LLM with retrieval over curated medical resources and auxiliary tools (a browser and calculators). A panel of board-certified clinicians compared it with ChatGPT-4, Bing, and Bard on factuality, completeness, user preference, and adversarial safety, and it showed significant improvement over the standalone LLMs. The evaluation, however, relies on expert-panel scoring rather than standardized quantitative benchmarks, does not report response latency or retrieval overhead, and offers no structured per-claim traceability linking each output statement to its source.

Xiong et al. [7] introduced the MIRAGE benchmark and systematically evaluated RAG configurations (retrievers, corpora, and chunk sizes) on MedQA, PubMedQA, and other medical question-answering datasets, establishing practical guidance for medical RAG design. The study focuses on retrieval configuration and accuracy alone. It does not evaluate explainability, source traceability, or hallucination-specific faithfulness metrics, and it does not benchmark end-to-end latency against a non-RAG baseline.

D. Reliability, Domain-Specific Systems, and Hallucination Mitigation

Vladika et al. [8] decomposed LLM-generated answers into atomic facts and verified each fact independently against a curated medical-guideline knowledge base, improving both factual accuracy and fact-level traceability. The framework, however, is evaluated only on long-form question answering rather than on diagnostic tasks, and it does not benchmark the reliability–efficiency trade-off (for example, response latency) against a standalone LLM baseline.

Luo et al. [9] developed a domain-specific RAG pipeline for ophthalmology that grounds LLM responses in curated clinical guidelines and improves accuracy over a non-RAG baseline. The system is narrowly scoped to a single medical specialty, and its evaluation includes neither explainability scoring nor a systematic hallucination/faithfulness metric. This limits its generalizability to a broader diagnostic pipeline.

Xu et al. [10] proposed MEGA-RAG, which adds a multi-evidence refinement step that cross-checks retrieved evidence before answer generation, targeting hallucination reduction in public-health LLM applications. The work centers on hallucination mitigation alone. It does not address source traceability or explainability for clinicians, and it does not benchmark diagnostic accuracy or response latency against a standalone baseline.

Table 1
Summary of Reviewed Studies and the Limitation Relevant to This Work

Study	Main Contribution	Limitation Relevant to This Work
Pinna et al. [1]	RAG differential-diagnosis assistant using PubMed abstracts; rated more trustworthy than plain ChatGPT-4	Compares exposition clarity; no systematic benchmark of hallucination, accuracy, or latency
He et al. [2]	Survey of biomedical RAG; “biomedical RAG trilemma” (reasoning depth, latency, privacy)	Survey only; no implemented or validated diagnostic pipeline
Liang et al. [3]	RGAR: retrieval of factual (EHR) and conceptual knowledge	No explainability or traceability; latency not evaluated
Zhao et al. [4]	MedRAG: four-tier diagnostic knowledge graph to reduce misdiagnosis	No joint benchmark against a non-RAG LLM; no structured clinician evaluation of explainability
Ansari et al. [5]	QLoRA-fine-tuned Llama 3.2-3B with RAG on hospital data	Focused on deployment efficiency; no traceability to source evidence
Zakka et al. [6]	Almanac: retrieval over curated resources plus tools; clinician-panel evaluation	Expert-panel scoring only; no latency; no per-claim traceability
Xiong et al. [7]	MIRAGE benchmark; evaluation of retrievers, corpora, and chunk sizes	No explainability, traceability, or faithfulness metrics; no end-to-end latency comparison
Vladika et al. [8]	Atomic fact checking against a guideline knowledge base	Long-form QA only; no reliability–efficiency trade-off against a baseline
Luo et al. [9]	Ophthalmology-specific RAG grounded in clinical guidelines	Single specialty; no explainability scoring or systematic faithfulness metric
Xu et al. [10]	MEGA-RAG: multi-evidence refinement to mitigate hallucination in public health	Hallucination only; no traceability, diagnostic accuracy, or latency benchmark

III. RESEARCH GAP AND OBJECTIVES

Across the ten reviewed studies, one gap recurs. No existing system integrates grounded retrieval, per-claim traceability, and a rigorous quantitative comparison of accuracy, reliability, and efficiency against a standalone LLM baseline within a single diagnostic pipeline. This gap motivates the proposed architecture. The research objectives are as follows.

1. To design and develop a RAG framework that grounds diagnostic suggestions in verified medical sources.
2. To implement a traceable and explainable diagnostic pipeline in which every suggestion links back to its supporting evidence.
3. To evaluate the accuracy, reliability, and efficiency of the proposed RAG–LLM system against a standalone (non-RAG) LLM baseline.

IV. PROPOSED FRAMEWORK AND PLANNED EVALUATION

A. Research Type and Approach

The study is categorized as applied research, since it addresses a practical problem rather than developing a new theoretical model. The individual components (LLMs, retrieval mechanisms, and medical knowledge bases) already exist in the literature; this research integrates them into a unified, traceable diagnostic pipeline. The study follows a quantitative approach: the effectiveness of the architecture will be assessed with measurable experimental metrics, namely diagnostic accuracy, hallucination/faithfulness rate, response latency, retrieval-time overhead, and explainability rating (a structured rubric or Likert scale).

B. Research Design

The study adopts an experimental research design. The proposed RAG–LLM architecture will be implemented and evaluated under different configurations, with and without retrieval grounding, and its performance will be compared with a standalone LLM baseline using predefined quantitative metrics. The design combines three complementary strands:

- an exploratory phase, which reviews the literature on RAG, medical LLMs, and explainable clinical AI in order to select an LLM backbone, a retrieval technique, and a knowledge base;
- a descriptive phase, which documents the system architecture (the retrieval, augmentation, generation, and traceability layers) and how its components interact; and
- an experimental phase, which implements and evaluates the complete system on a benchmark diagnostic dataset.

C. Proposed Architecture

The proposed pipeline has four layers, shown in Fig. 1. The retrieval layer draws on credible medical knowledge sources. The augmentation layer injects the retrieved evidence into the LLM’s context. The

generation layer produces the diagnostic suggestion. The traceability layer attributes each output claim to its supporting source. The standalone LLM baseline uses the same underlying model without the retrieval and augmentation steps.

D. Research Methodology: Seven Phases

The methodology consists of seven phases.

1. Literature review and research-gap identification. A systematic review of recent RAG, medical-LLM, and explainable clinical decision-support literature establishes the research gap.
2. Objectives, research questions, and hypotheses. The gap is translated into a primary research question and sub-questions, each paired with a testable null/alternative hypothesis (H0/H1) covering accuracy, reliability, explainability, and efficiency.
3. Selection of knowledge base, dataset, and tools. This phase selects the medical knowledge base for retrieval (for example, PubMed abstracts, clinical guidelines, or ICD/SNOMED references), the benchmark evaluation dataset (for example, MedQA or PubMedQA), and the LLM backbone, embedding model, vector database, and orchestration framework.
4. Design of the RAG–LLM architecture. The retrieval layer, the augmentation layer that injects retrieved evidence into the LLM’s context, the generation layer, and the traceability layer that attributes each output claim to its source are integrated.
5. System implementation. Both the standalone LLM baseline and the full RAG–LLM pipeline are implemented on the same underlying LLM, so that any measured performance difference can be attributed specifically to the retrieval-augmentation mechanism.
6. Experimental evaluation. The systems are evaluated quantitatively using diagnostic accuracy, hallucination/faithfulness rate, explainability rating (through a clinician/expert rubric), and response latency.
7. Analysis and validation. Results are analyzed against the standalone baseline to test hypotheses H1–H4, covering diagnostic accuracy, reliability, explainability, and the efficiency trade-off introduced by retrieval.

E. Evaluation Metrics and Hypotheses

Table 2 lists the four hypotheses and the metrics that will be used to test them.

Table 2
Planned Hypotheses and Evaluation Metrics

ID	Dimension	Planned Metric
H1	Diagnostic accuracy	Accuracy on the benchmark dataset
H2	Hallucination/faithfulness	Rate of unsupported or incorrect claims
H3	Explainability	Clinician/expert rubric rating (Likert scale)
H4	Efficiency	Response latency and retrieval overhead

F. Reproducibility

To support reproducibility, the complete implementation will be thoroughly documented. This covers the architecture, knowledge-base sources, retrieval configuration (chunk size, top-k value, and vector database), model-selection criteria, and experimental setup. The implementation will rely on publicly available LLMs, open-source retrieval frameworks, and standard software libraries wherever possible. Random seeds used during testing will be recorded, and experiments will be run multiple times with averaged results so that a single anomalous run does not misrepresent system performance. System code, configuration files, and, where licensing permits, knowledge-base construction scripts will be made publicly available with setup instructions.

V. SYSTEM IMPLEMENTATION

A prototype of the proposed pipeline, named RAG-MedDx, was implemented as a simplified, offline simulation of the architecture of Fig. 1. The application runs entirely in a web page on the local device: it accepts a free-text patient case, retrieves reference material from a small local knowledge base, and generates a citation-traced suggestion without calling any external model or cloud API. The configuration of the prototype is summarized in Table 3.

Table 3
Configuration of the RAG-MedDx Prototype

Component	Setting
Interface	Browser-based page that runs locally on the device
Knowledge base	Six locally stored reference entries, each tagged with its textbook source (for example, Braunwald's Heart Disease and Goldman-Cecil Medicine)
Retrieval engine	Keyword matching between the query terms and the stored entries
Generation mode	Offline, rule-based simulated answer generation (a stand-in for an LLM)
Citations	Inline numeric markers that link each statement to a retrieved source
Cloud API	Disabled; no external model or API is called
No-match handling	A message is shown instead of a generated answer
Displayed measurements	Response time, number of sources retrieved, number of query terms matched

A. Pipeline

The prototype processes a case in four steps that are shown in the pipeline panel of its interface: parsing the query, retrieving sources, generating the answer, and attaching citations. These steps correspond to the retrieval, generation, and traceability layers of Fig. 1.

B. Retrieval and Answer Generation

The query is split into terms, and the terms are matched against the local knowledge base. Every retrieved source is displayed with a citation tag, its textbook reference, and the terms that matched the query. The answer is composed by simple rules from the retrieved sources, and each statement carries an inline citation marker that points to the source it is based on. When no source matches, no answer is generated and the user is asked to describe the symptoms more specifically.

C. Interface and Status Display

A status panel shows the number of loaded documents, the retrieval engine, the generation mode, and the state of the cloud API. After every query the interface reports the response time, the number of sources retrieved, and the number of query terms matched. The interface also states that the system is a simplified offline simulation for research and demonstration purposes, and that it is not a diagnostic tool and not for clinical use.

D. Framework for Quantitative Evaluation

For the quantitative comparison with a standalone LLM described in Section IV, a separate Python evaluation framework was written. It contains the retrieval, augmentation, generation, and traceability

modules, the baseline, and the metric computation of Table 2. Its default configuration is as follows. Documents are split into overlapping chunks of 120 words with an overlap of 20 words, and the four most similar chunks are retrieved for each question using TF-IDF (unigrams and bigrams) with cosine similarity; dense sentence embeddings are available as an alternative retriever. The baseline and the RAG-LLM system use the same LLM, the same temperature setting, and the same JSON output schema (an answer and a list of short claims), so that only retrieval differs. In the RAG-LLM system each claim must cite the labels of the evidence chunks that support it; citations to labels that were not retrieved are discarded in the traceability layer and counted separately. Unparseable answers are counted as incorrect. Accuracy of the two systems is compared on identical questions with an exact McNemar test, and results are averaged over several random seeds. This framework has not yet been run on a benchmark dataset, and no results obtained with it are reported in this paper.

The correctness of the framework itself was checked with an automated offline test. The test covers chunking, parsing of model output, retrieval, citation validation, the McNemar computation, and a complete run of both systems on a small built-in demonstration set using a deterministic mock model in place of an LLM. The test passed. Because the mock model performs no medical reasoning, this test verifies only that the components work together as intended; the figures it prints are not measurements of diagnostic performance and are not reported as results.

VI. RESULTS AND DISCUSSION

The developed prototype, RAG-MedDx, was implemented as a simplified, offline simulation of a Retrieval-Augmented Generation (RAG) pipeline for medical diagnosis support. The system retrieves relevant reference material from a small local knowledge base in response to a patient case query, then generates a citation-traced diagnostic suggestion without relying on a cloud-based large language model. The results in Sections VI-A to VI-D are demonstrations on sample queries. Accuracy, hallucination rate, and a comparison with a standalone LLM have not yet been measured.

A. Offline Retrieval Pipeline

The developed application accepts a free-text patient case description and processes it entirely within the local environment, without any external API or cloud-based model call. The interface displays the system status, confirming that the cloud API is disabled and that generation runs in offline/simulated mode.

B. Retrieval and Citation Traceability

For the sample patient case query “chest pain radiating to left arm,” the system retrieved the most relevant reference sources from its local knowledge base using keyword-based matching. Each retrieved source is displayed with a citation tag and the specific terms that matched the query, allowing the retrieval step to be traced and verified.

The generated answer references the retrieved sources directly using inline citation markers, so that each claim in the output can be traced back to a specific source document.

C. Handling of Unmatched Queries

Query handling was tested using a case with no meaningful keyword overlap with the local knowledge base. The test query was the single misspelled word “Swetting,” which matches no term of the knowledge base. In this case, the system correctly reported that no close keyword matches were found, rather than generating an unsupported or fabricated suggestion.

D. Response Generation Performance

The prototype measured the response time for each simulated retrieval-and-generation cycle. For the sample query above, the system reported a response time of 0.77 seconds, with 2 sources retrieved and 4 query terms matched. For the unmatched query of Fig. 5, the response time was 0.49 seconds, with 0 sources retrieved and 0 query terms matched.

E. Discussion

The developed prototype demonstrates the feasibility of a Retrieval-Augmented Generation approach to medical diagnosis support in which every generated suggestion is grounded in explicit, retrievable source material. Rather than presenting a diagnostic suggestion as an opaque model output, the system exposes the specific reference documents and matched terms that informed the response.

This is intended to address the two problem areas motivating the research: the hallucination and reliability gap in LLM-generated medical text, and the lack of explainability and traceability in existing diagnostic tools. By requiring every generated statement to carry an inline citation back to a retrieved source, the architecture makes the reasoning path visible and auditable, rather than trusting a generated answer directly.

The offline, simulated design also supports a resource-efficient and privacy-preserving deployment model: no patient case data leaves the local environment, and no dependency on a cloud-based large language model is required for the demonstration to function. The keyword-based retrieval used here is a simplified stand-in for a full embedding-based retrieval system, and the rule-based answer generation is a simplified stand-in for a fine-tuned or prompted LLM; both choices keep the prototype fully offline and inspectable while illustrating the intended pipeline structure.

The handling of unmatched queries further illustrates a core reliability goal of the framework: rather than generating a plausible-sounding but ungrounded answer when no relevant source material is found, the system explicitly reports the absence of a match. This behavior is intended to reduce the risk of hallucinated diagnostic suggestions.

Overall, the implementation demonstrates that retrieval traceability, citation-grounded generation, and explicit handling of low-confidence cases can be integrated into a single, inspectable pipeline. These observations are qualitative. The sample retrieval also shows a limit of keyword matching: the pneumonia source was retrieved only through the term “chest pain,” so the retrieved set contains a source that is less relevant to pain radiating to the left arm. The effect of the approach on diagnostic accuracy, hallucination rate, and response time compared with a standalone LLM has not been measured.

Beyond the prototype, the Python evaluation framework was designed so that the comparison with a standalone LLM isolates the effect of retrieval. The baseline and the RAG–LLM system use the same model, the same decoding settings, and the same output schema, and they are run on identical questions, so that differences in accuracy can be attributed to the retrieved evidence and tested with a paired McNemar test. Each claim in the RAG–LLM output must cite the retrieved chunks that support it, and citations to chunks that were not retrieved are discarded and counted, which prevents invented references from appearing to provide traceability. Two measures complement each other in this design: the hallucination rate checks claims against reference evidence, while the citation-support rate checks whether each claim is supported by the specific chunk it cites. Several design choices will affect how the future results should be read. PubMedQA contains the gold abstract of every question in the knowledge base, which favors retrieval; PubMedQA answers are yes, no, or maybe rather than free-form diagnoses; and an automatic claim judge can be biased or noisy. These points will be reported together with the measured values rather than after them.

F. Planned Quantitative Evaluation

The demonstrations above do not measure accuracy, hallucination rate, or explainability, and they do not include a baseline. The planned evaluation of Section IV will compare the RAG–LLM system with a standalone LLM on the same randomly drawn subsets of questions from a benchmark dataset (PubMedQA in the evaluation framework; MedQA-style multiple-choice questions can be loaded as well). Experiments will be repeated for several random seeds, and the mean and standard deviation over the seeds will be reported. Latency will be measured on a fixed hardware configuration, which will be stated together with the results, and explainability will be rated by clinicians or experts using a structured rubric. No results table is included because this evaluation has not been carried out.

VII. LIMITATIONS

The present paper has the following limitations.

- The demonstrated prototype, RAG-MedDx, is a simplified offline simulation. It uses keyword matching over six local documents and rule-based answer generation instead of an LLM, and its outputs have not been clinically validated.

- No quantitative results (accuracy, hallucination/faithfulness rate, explainability rating) and no comparison with a standalone LLM are reported. The only measurements are the response times of two sample queries (0.77 s and 0.49 s), measured on a single device. All other statements about performance are expectations that remain to be tested.
- Keyword matching depends on exact term overlap: a misspelled query retrieves nothing, and a source can be retrieved through a single shared term.
- The Python evaluation framework has passed an offline functional test with a mock model, but it has not yet been run with a real LLM on a benchmark dataset. The LLM backbone for the final evaluation has not been fixed, and the planned PubMedQA experiment uses a knowledge base that contains the gold abstract of every question, which makes retrieval easier than in real clinical use. PubMedQA is also a yes/no/maybe question-answering task rather than free-form diagnosis. If an LLM is used to judge whether claims are supported, its judgments may be biased or noisy, and the judge used will be reported with the results.
- The literature review covers ten recent studies and is not claimed to be exhaustive.
- Planned explainability ratings depend on clinician or expert judgment through a structured rubric, which introduces subjectivity, and the automatic claim-support judgement can itself be noisy.

VIII. CONCLUSION

This paper has presented RAG-MedDx, a preliminary offline prototype of a RAG-based framework for medical diagnosis support, together with the plan for testing it against a standalone LLM. A review of ten recent studies identifies a consistent gap: existing medical RAG systems address retrieval technique, accuracy, or explainability in isolation, and none of the reviewed studies combines grounded retrieval, per-claim traceability, and a quantitative comparison against a standalone LLM baseline in one pipeline. The proposed architecture has retrieval, augmentation, generation, and traceability layers. The prototype demonstrated citation-traced answers, explicit handling of a query without matching sources, and response times of 0.77 s and 0.49 s for two sample queries. A Python evaluation framework implementing the baseline, the RAG–LLM pipeline, citation validation, and the metrics for hypotheses H1–H4 was also built and passed an offline functional test. These demonstrations are qualitative or functional, and no claim about improved accuracy, reduced hallucination, or explainability is made in this paper. Future work will run the framework with a real LLM on PubMedQA, collect blinded expert ratings of explainability, report the resulting metrics with their variation over random seeds, replace the keyword matching and rule-based generation of the prototype with embedding-based retrieval and an LLM, and refine the architecture in light of the findings.

Declarations

Acknowledgment

The author gratefully acknowledges the Department of Artificial Intelligence and Data Science, Sanjivani University, Kopergaon, Maharashtra, for the guidance and support extended during this research. The author also thanks the faculty guide and peer reviewers for their valuable feedback on the research design and methodology presented in this paper.

References

1. Pinna S, Massa SM, Fenu M, Casti G, Riboni D (2025) Integration of retrieval-augmented generation technique for LLM-based differential diagnosis assistant, in Proc. 18th ACM Int. Conf. Pervasive Technologies Related to Assistive Environments (PETRA '25), Corfu Island, Greece. 10.1145/3733155.3733192
2. He J, Zhang B, Rouhizadeh H, Chen Y, Yang R, Lu J, Chen X, Liu N, Teodoro D Retrieval-augmented generation in biomedicine: A survey of technologies, datasets, and clinical applications, arXiv preprint arXiv:2505.01146, 2025.
3. Liang S, Zhang L, Zhu H, Wang W, He Y, Zhou D (2025) RGAR: Recurrence generation-augmented retrieval for factual-aware medical question answering, in Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, pp. 4006–4033, arXiv:2502.13361
4. Zhao X, Liu S, Yang S-Y, Miao C MedRAG: Enhancing retrieval-augmented generation with knowledge graph-elicited reasoning for healthcare copilot, in Proc. ACM Web Conf. 2025 (WWW '25), 2025, pp. 4442–4457. 10.1145/3696410.3714782
5. Ansari MS, Khan MSA, Revankar S, Varma A, Mokhade AS (2025) Lightweight clinical decision support system using QLoRA-fine-tuned LLMs and retrieval-augmented generation, arXiv preprint arXiv:2505.03406
6. Zakka C, Shad R, Chaurasia A, Dalal AR, Kim JL, Moor M, Fong R, Phillips C, Alexander K, Ashley E et al (2024) Almanac—Retrieval-augmented language models for clinical medicine. NEJM AI 1(2). 10.1056/Aloa2300068
7. Xiong G, Jin Q, Lu Z, Zhang A Benchmarking retrieval-augmented generation for medicine, in Findings of the Association for Computational Linguistics: ACL 2024, 2024, pp. 6233–6251. 10.18653/v1/2024.findings-acl.372
8. Vladika J et al (2025) Improving reliability and explainability of medical question answering through atomic fact checking in retrieval-augmented LLMs. arXiv preprint arXiv:2505.24830
9. Luo MJ, Pang J, Bi S, Lai Y, Zhao J, Shang Y et al (2024) Development and evaluation of a retrieval-augmented large language model framework for ophthalmology. JAMA Ophthalmol 142(9):798–805. 10.1001/jamaophthalmol.2024.2513
10. Xu S, Yan Z, Dai C, Wu F (2025) MEGA-RAG: A retrieval-augmented generation framework with multi-evidence guided answer refinement for mitigating hallucinations of LLMs in public health, Front. Public Health, vol. 13, art. 1635381. 10.3389/fpubh.2025.1635381

Figures

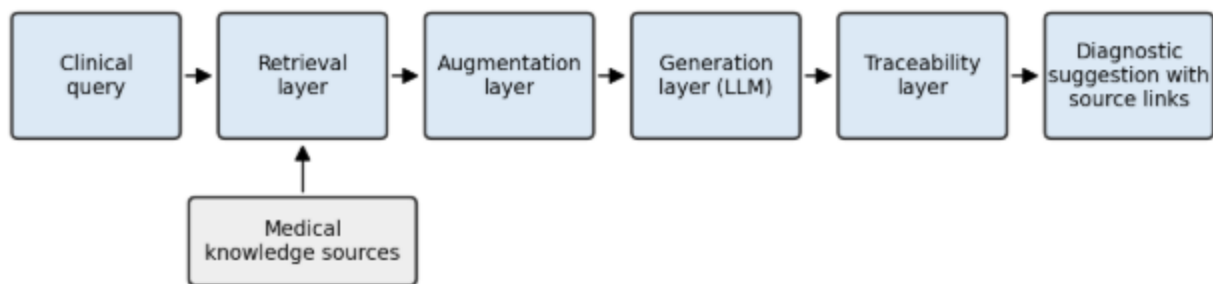


Figure 1

Proposed RAG-LLM diagnostic pipeline.

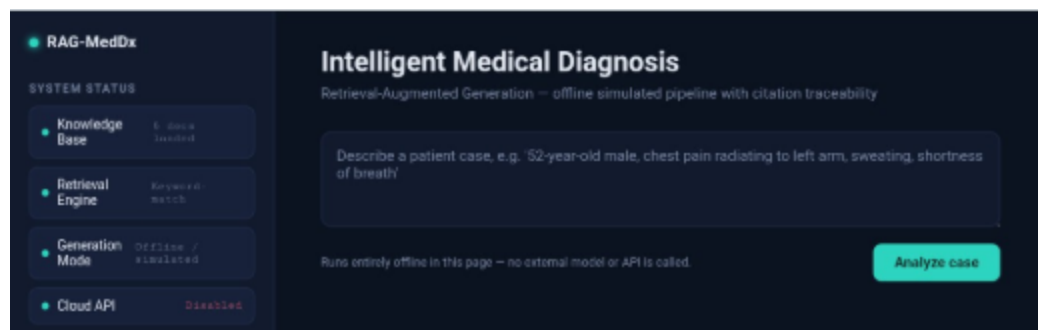


Figure 2

Working interface of the RAG-MedDx system.

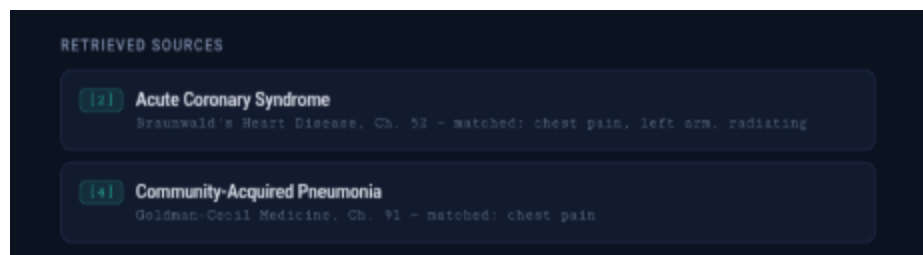


Figure 3

Retrieved sources with citation tags for a sample query.

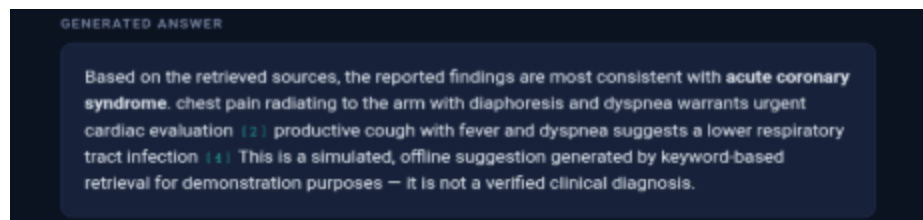


Figure 4

Generated answer with inline citation traceability.

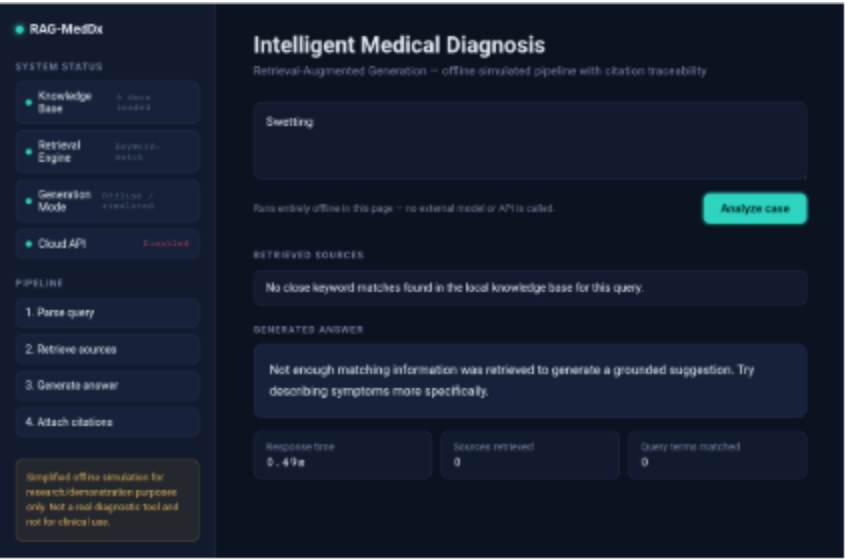


Figure 5

Handling of a query with no matching sources.

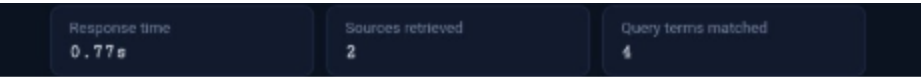


Figure 6

Response generation performance for a sample query.