Noise, Distraction, and Mitigation: An Analysis of RAG Failure Modes in Medical Question Answering

Md Towhidul Islam Rahat
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
towhidul.rahat@northsouth.edu

Riad Safowan
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
riad.safowan@northsouth.edu

Mirza Abir
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
mirza.abir@northsouth.edu

Rashedur M. Rahman
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
rashedur.rahman@northsouth.edu

***Abstract*—Retrieval Augmented Generation (RAG) is an idea to give Large Language Models (LLMs) a much needed boost by tapping into external knowledge sources. However, there is still much to learn about how well it works in the field of medicine. At present, we do not have enough information to say for certain. In this research, we take a top-of-the-line medical LLM called MMed-Llama-3-8B and test on a 500 question challenge called the PubMedQA benchmark. We found a significant performance degradation when integrating RAG system to the model. When the RAG model provides additional documents, the baseline accuracy of 68.8% has been dropped to as low as 17.6%. To investigate the source of error, we analyzed 165 cases in detail where the performance was degraded by RAG system. The primary issues we found were that RAG retrieved some documents that diverted the model from the context in 41.8% of cases, whereas in 37.6% cases the retrieved documents contradicted knowledge. To mitigate this, we devised a better way to prompt the model that could improve the performance of the RAG setup to 59.6% a 186% boost over just using RAG as it is. Our research reveals the fact that within domains such as medicine, the mere application of RAG is inadequate and the models should be able to manage noise that are inherent in data from external sources.**

***Index Terms*—Retrieval-Augmented Generation, Large Language Models, Medical Question Answering, Prompt Engineering, RAG Failure Modes**

I. Introduction

In the field of Medicine, Large Language Models (LLMs) show promising results in tasks such as retrieving clinical information, understanding medical documents, and diagnosing problems in general [1], [2]. However, these LLM models fall short on remaining up to date as newer research appears and the models knowledge-base stale. This is a major concern, especially in healthcare, where even minor errors can be detrimental to patient health.

Retrieval-Augmented Generation (RAG) techniques can be applied to address this issue. It empowers LLMs to gather external evidence during the inference step [3], [4]. This approach shows an improvement in LLM's ability to acquire factually correct responses on general knowledge questions. However, when it comes to the more complex world of medicine, the behavior of the RAG is still unpredictable, especially when dealing with noisy information.

To investigate the reliability of medical RAG systems, we evaluate a state-of-the-art medical LLM-MMed-Llama-3-8B on the PubMedQA data set [5] using a standard dense retrieval pipeline. We find that naive RAG significantly harms clinical response precision, dropping from a baseline of 68.8% golden-context baseline to as low as 17.6% as additional retrieved texts are provided. This failure, however, is not random, and we show that it is systematically caused by *distracting* or *contradictory* retrieved content.

To mitigate the problem, we closely examined 165 cases where the RAG system failed. Additionally, we developed a more robust training idea that helps the model ignore misleading information, resulting in performance improvements when more documents were added, with an accuracy of 59.6%, an 186% gain over naive RAG system.

This paper makes four primary contributions:

- We provide the first systematic analysis of RAG failure modes in medical question answering.
- Identifying distracting and contradictory retrieval as the primary causes of degradation.
- We demonstrate that naive RAG is not plug-and-play in medical contexts and can perform significantly worse than no retrieval.
- We introduce a robust few-shot prompting strategy that substantially improves resilience to retrieval noise and recovers high-performance RAG behavior.

We show that improving retrieval alone is not enough; robust generation is important in specialized, safety-critical domains.

II. Related Work

A. RAG Systems

Retrieval-Augmented Generation (RAG) was initially developed to enhance language models by conditioning their output on dynamically retrieved evidence from the web [3]. Since then, researchers have shown that it is quite useful across a range of tricky NLP tasks, such as, asking a computer to come up with answers to questions on the fly and fact checking [4], [13], [14]. The fundamental premise is that as the model collects more context from the evidence, the more reliable its answers become. However, recent studies have found contradictory results. While retrieving information from the web, models often end up generating wrong response especially when the data-sources are poorly formed or less relevant. [12], [18]. Most recent researches often prioritize measuring model performance by analyzing good responses, while overlooking the quality of malformed reactions due to poor information sources. As a result, we still lack a clear understanding of how well RAG models perform when tackling real-world problems with the available information at their disposal.

B. Medical Question Answering

Biomedical QA has really taken big leaps, thanks to the availability of specialized datasets and model architectures that are specifically designed to work in the medical field. One widely used benchmark for testing how well clinical reasoning skills can be applied to scientific literature is PubMedQA [5]. We see somewhat impressive results from models like Med-PaLM and DoctorGLM [1], [16] under controlled conditions. These models excel in strength with parametric knowledge. However, the model's knowledge can become outdated as newer research comes in. Some newer RAG systems are trying to address this with an approach called Corpus-Grounded Inference [6], [15]. However, the testing often takes place in easily manageable scenarios and usually assumes that the model can fairly easily find the evidence needed to answer the question. Real-world situations are much more complex in the face of conflicting evidence, other comorbidities, and relevant yet unclear texts.

C. Prompt Engineering for Robustness

Prompt engineering has proven effective for enhancing model controllability, reasoning, and adherence to output constraints [10], [11]. In the context of RAG, recent work has observed that instruction-based or example-driven prompting can reduce the impact of noisy retrieval signals [12]. However, current approaches lack domain-aware analysis explaining *why* particular prompting strategies succeed or fail when exposed to conflicting biomedical evidence. Furthermore, prompt optimization techniques remain largely evaluated in general-purpose scenarios that do not require strict factual fidelity or risk-aware handling of contradictory information.

D. Medical RAG and Rationale-Guided Retrieval

Recent work has extended RAG to the biomedical domain, where the quality of reliable evidence is critical. Rationale-guided RAG systems improve retrieval precision by explicitly identifying clinically meaningful explanatory evidence rather than relying on surface-level similarity. For example, Sohn et al. [25] show that incorporating structured rationales into retrieval enhances grounding and improves answer reliability in medical QA. However, these approaches mainly strengthen retrieval, while our work addresses a complementary challenge. Even with improved retrieval, biomedical literature often contains ambiguity, conflicting findings, and partially relevant content

III. Methodology

We design a four-phase experimental pipeline aligned with three research questions (RQ1-RQ3) to investigate the reliability of RAG in medical question answering.

A. Experimental Setup

Model: Our baseline generative model was MMed-Llama-3-8B [9], a state-of-the-art LLM pre-trained for the medical domain. All experiments were run on this model to ensure controlled comparisons.

Evaluation Dataset: We used the PubMedQA dataset [5]. To generate the test set, we sample randomly 500 questions from the pqa labeled split with a random seed (42) so that our results are reproducible.

Knowledge Base: To construct a realistic knowledge base for RAG model we used pqa unlabeled split and create a corpus of 61,249 medical abstracts.

B. Baseline Experiments

To understand the model's performance boundaries, we establish two key baselines described below.

No-Context Baseline: We evaluated the model on the 500 questions with a zero-shot prompt where no supporting context is provided, so only the question is provided.

Golden-Context Baseline: In this case, we evaluated the model with human-annotated "golden" context that is provided with the PubMedQA dataset. With this golden text the retriever achieves the perfect precision and recall.

C. RAG Pipeline Implementation

We implemented a standard RAG pipeline using the following components::

Embedding Model: Our corpus contains 61, 249 documents. We used with all-MiniLM-L6-v2 sentence transformer [7] to embed those documents

We select this widely-adopted embedding model to reflect a common baseline RAG configuration, enabling our findings to generalize to typical deployment scenarios. While more recent encoders exist, our focus is on analyzing *failure modes* and *generator robustness* rather than optimizing retrieval quality, making this standard baseline appropriate for our research objectives.

Vector Store: We index the embeddings using FAISS [8], which increases efficiency in the search for similarity for the retrieval process.

Retrieval Strategy: For retrieval, we embed the query first and then use cosine similarity to retrieve the top k-most documents.

This pipeline matches widely-used RAG implementations in both research and production.

D. Phase 1: Impact of Context Quantity (RQ1)

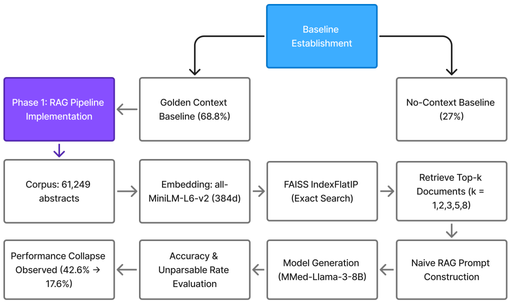
Baseline Establishment
Phase 1: RAG Pipeline Implementation
Golden Context Baseline (68.8%)
No-Context Baseline (27%)
Corpus: 61,249 abstracts
Embedding: all-MiniLM-L6-v2 (384d)
FAISS IndexFlatIP (Exact Search)
Retrieve Top-k Documents (k = 1,2,3,5,8)
Performance Collapse Observed (42.6% → 17.6%)
Accuracy & Unparsable Rate Evaluation
Model Generation (MMed-Llama-3-8B)
Naive RAG Prompt Construction

Fig. 1. RAG Pipeline Phases 1 & 2.

Using a naive prompt template, we perform a systematic *k-sweep* evaluation. By varying the values of k ($k \in \{1, 2, 3, 5, 8\}$) we measure the impact of quantity of context on the precision of the model and identify any performance trends.

E. Phase 2: Failure Mode Analysis (RQ2)

We take our best basic RAG setup using a single retrieved document, which was able to correctly answer 42.6% of the questions. Then we compare it to the outcome, when the model was provided with the perfect context (the ”Golden Context”), which produced 68.8% correct response. The questionnaire consists of 165 questions on which the model performed better when given the right context. However, the model's performance degraded when the RAG was allowed to fetch the context on its own. This degradation in performance leads to the primary conclusion that the retrieval step adversely impacted the models performance.

To investigate the reasons for degraded performance, we followed two approaches:

Failure Isolation: We isolated the cases where the retrieved content actively impacted the model's answer, and excluded those cases where the model's incorrect responses can be attributed to model quirkiness. We then manually analyzed each individual case, looking for patterns to understand what went wrong with the retrieved context.

Taxonomy Construction: We followed a programmatic approach to separate the failure cases into four broad categories, a) The retrieved text contains information that is directly contradictory to the correct response, b) The retrieved text is ambiguous and vague leading the model to chose incorrect responses c) The retrieved document is entirely unrelated to the question the model is tasked to answer, and d) The model failed to parse any answer from the retrieved context.

F. Phase 3: Mitigation Strategy (RQ3)

Guided by our error taxonomy, we developed a robust few-shot prompt inspired by in-context learning principles [10]. The prompt:

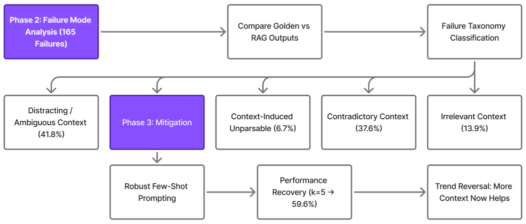
Phase 2: Failure Mode Analysis (165 Failures)
Compare Golden vs RAG Outputs
Failure Taxonomy Classification
Distracting / Ambiguous Context (41.8%)
Phase 3: Mitigation
Context-Induced Unparsable (6.7%)
Contradictory Context (37.6%)
Irrelevant Context (13.9%)
Robust Few-Shot Prompting
Performance Recovery (k=5 → 59.6%)
Trend Reversal: More Context Now Helps

Fig. 2. RAG Pipeline Phases 3 & 4.

- Demonstrated explicit rejection of contradictory evidence
- Highlighted recognition of answer-bearing context
- Reinforced strict label compliance (*yes/no/maybe*)

We re-evaluated performance at $k = 1$ and $k = 5$ to assess whether generator-side robustness can counter retrieval noise and restore the benefits of additional context.

IV. Results

A. Baseline Performance

Our baselines establish the parametric and oracle-assisted performance bounds of the system. The No-Context condition achieved an accuracy of **27.0%**, reflecting the model's limited factual recall for specialized clinical knowledge. Providing the human-annotated golden context increased accuracy to **68.8%**, confirming that the model is capable of producing clinically correct answers when supplied with precise, answer-bearing evidence.

B. RQ1: Naive RAG Degrades Performance

Fig. 3 illustrates the central finding of Phase 1: introducing retrieved context does not guarantee performance gains. Instead, naive RAG consistently reduced correctness.

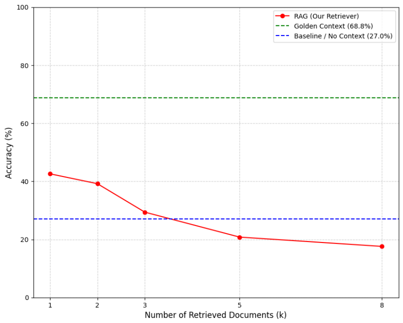

Fig. 3. Answer accuracy as a function of retrieved document count (k) under naive prompting. Performance monotonically decreases as retrieval quantity increases.

At $k = 1$, accuracy dropped to **42.6%**, already substantially below the golden-context condition. As k increased, performance progressively collapsed, falling to **20.8%** at $k = 5$ and **17.6%** at $k = 8$. This represents a performance level significantly worse than having no context at all (27.0%).

The accuracy drop is directly correlated with a rise in unparsable responses. Fig. 4 shows a strong positive correlation between retrieval . The No-Context baseline had a high unparsable rate (26.2%), which dropped to just 2.4% for the clean Golden Context run. With our naive RAG, the unparsable rate increased with k, reaching 59.6% at k=8. This proves that the model becomes “overwhelmed” by noisy context and fails to follow basic formatting instructions.

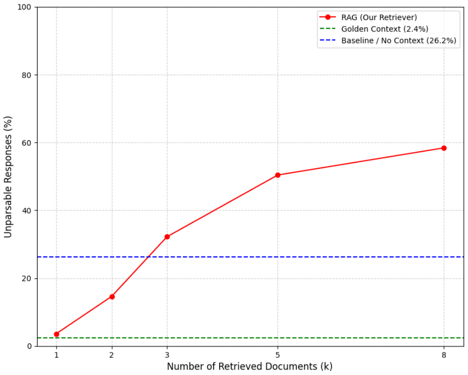

Fig. 4. Unparsable response rate under naive RAG. Higher k values increase output instability and response formatting failures.

These findings demonstrate that naive RAG introduces *cascading error propagation*, where increased retrieval leads to increased formatting failure, contributing to accuracy degradation.

C. RQ2: Failures are Systematic and Misleading

We investigated the **165** RAG-induced failures identified by comparing the golden-context and naive RAG ($k = 1$) runs. The failures followed structured, non-random patterns.

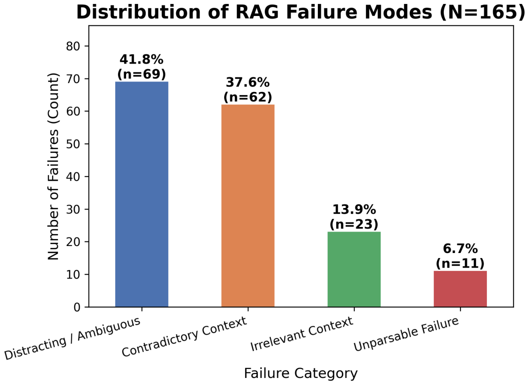

Fig. 5. Distribution of RAG-induced failure modes for the 165 analyzed errors. Distracting and contradictory contexts drive the majority of failures.

As shown in Fig. 5, two failure types dominate:

Distracting/Ambiguous Context (41.8%): Retrieved passages were topically coherent but non–answer-bearing, often causing the model to hedge (producing “maybe” inaccurately) or infer incorrect conclusions.

Contradictory Context (37.6%): Retrieved evidence explicitly conflicted with the ground-truth label, misleading the model toward incorrect answers.

The remaining failures comprised:

- **Irrelevant context** (13.9%): No useful signal for answering
- **Context-induced unparsable responses** (6.7%): Format breakdown due to heterogeneous evidence

Interpretation: The retriever is **semantically aligned but not answer-aware**. Contextual overlap alone is insufficient for reliable retrieval in specialized medical settings.

D. RQ3: Robust Prompting Restores RAG Benefits

We tested whether generator-side robustness could counter retrieval noise. Fig. 6 shows that our robust few-shot prompt substantially improved performance.

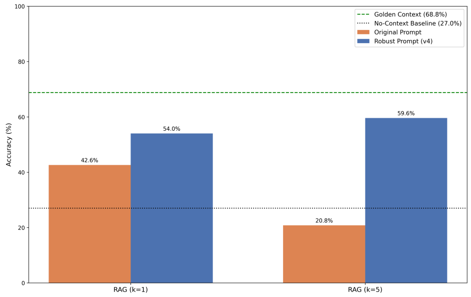

Fig. 6. Robust prompting significantly improves accuracy, particularly at higher k values, reversing the degradation trend observed with naive prompting.

Key improvements include:

- $k = 1$: **42.6%** → **54.0%** (+11.4 pp)
- $k = 5$: **20.8%** → **59.6%** (+38.8 pp)

Most importantly, performance trends inverted:

- Naive Prompt: $k \uparrow \Rightarrow$ Accuracy **down**
- Robust Prompt: $k \uparrow \Rightarrow$ Accuracy **up**

The robust prompting configuration at $k = 5$ reached **59.6% accuracy**, approaching the oracle upper limit (68.8%), and representing a **186% relative improvement** over the naive approach. This demonstrates that generator-side mitigation can recover the intended benefits of RAG, even under imperfect retrieval.

V. Discussion

A. Principal Findings

Our results demonstrate that RAG performance in medical question answering is not a fixed property of the architecture, but is highly dependent on the design of the generator prompt. An initial, naive interpretation of our findings might suggest that “more context is bad”: as the number of retrieved documents k increased under naive prompting, both accuracy and output parsability degraded sharply. However, our final experiments show that this view is incomplete.

A more precise conclusion is that *more context is harmful for a naive prompt, but beneficial for a robust prompt*. With naive prompting, increasing k from 1 to 5 reduced accuracy from 42.6% to 20.8%. In contrast, with our robust few-shot prompt, increasing k from 1 to 5 *improved* accuracy from 54.0% to 59.6%. This indicates that, when properly instructed, the model can locate the correct "needle" in a larger "haystack" of retrieved documents while ignoring noisy or misleading evidence.

B. Implications for RAG System Design

Our findings have important implications for RAG system design:

Retriever limitations are systematic. The failure-mode analysis reveals that standard dense retrievers are *recall-oriented but not answer-aware*. They reliably surface topically related passages but do not guarantee that the retrieved content supports the correct label. As a result, the generator is frequently exposed to distracting (41.8% of failures) and contradictory (37.6% of failures) evidence, which systematically degrades performance rather than improving it.

Generator robustness is critical. Focusing exclusively on improving retrieval quality is insufficient. Our results show that engineering the generator to be robust to noisy evidence is an equally powerful and more immediately accessible lever. A carefully designed robust prompt largely restored the benefits of retrieval and recovered 59.6% accuracy at $k = 5$, a 186% relative improvement over the naive RAG configuration. This suggests that RAG optimization should treat retrieval and generation as coupled components rather than optimizing the retriever in isolation.

Few-shot prompting is an effective robustness mechanism. Automated methods to discover robust prompts could make the techniques more accessible and scalable across different domains and tasks.

C. Limitations

The generalizability of our findings is limited by a number of constraints that are described below.

First, our experiments used a **single specialized model**, MMed-Llama-3-8B. Other architectures, model sizes, or training procedures may exhibit different robustness profiles. Larger general-purpose models may, for example, be more resilient to retrieval noise, potentially altering the magnitude or nature of the observed degradation.

Second, we focused exclusively on **one dataset and task formulation**: PubMedQA's yes/no/maybe classification questions. Many real-world medical applications require multi-step reasoning, longitudinal inference, or free-text clinical justification. Failure modes and mitigation strategies in these richer settings may differ from those observed here.

Third, our RAG pipeline employed a **single dense retriever configuration**: all-MiniLM-L6-v2 embeddings with FAISS-based nearest-neighbor search. Our conclusions therefore apply specifically to this widely used but relatively simple retrieval stack. Although more advanced retrieval architectures—including domain-adapted medical encoders, cross-encoder reranking, and hybrid sparse–dense methods—could reduce the prevalence of distracting or contradictory evidence, our choice of a standard baseline allows us to study failure modes that are likely present, to varying degrees, across many real-world RAG deployments.

Fourth, we followed a traditional iterative approach in designing prompts. we were able to make the model produce desirable outcome by adjusting our prompts based on model response. This is a time consuming process and often works better with appropriate domain knowledge. Similar prompt design might not work on different data-set or in different medical specialty. Using automated prompt optimization might make this step of the process more robust and scalable.

Fifth, our categorization of failure cases are purely based on established NLP and information retrieval concepts. However, without proper validation from domain experts this categorization approach might be vulnerable to wrongful labeling of cases.

Finally, we are testing this artificially. We assume that the system will simply grab all the documents at once and immediately deliver an answer. Nevertheless, actual clinical tools never work like that. They might ask follow-up questions, retrieve more information based on the first response, or have a human double-check things before finalizing. A lot of the failures we're seeing could probably be caught or fixed in those more interactive scenarios. Testing how these systems hold up when there is back-and-forth with users is something we definitely need to emphasize.

Going forward, we plan to test whether these same failure patterns show up when using the latest retrieval encoders and rerankers. Basically, we will see if even the best available technology today can sidestep these problems.

D. Future Work

This work points us a few things that are worth looking at in the future.

Evaluation with modern retrieval architectures: There are a few things we should look at next. The most obvious one is we need to see if these same problems show up when you use better retrieval systems. We have yet to determine if these limitations are a fundamental part of medical RAG. Or if they can be resolved using techniques like specialized medical embeddings, cross-encoder re-rankers, and hybrid search. Testing our failure categories and fixes against these newer methods would answer that. It would also tell us whether you still need to focus on making the language model itself more robust, even when the retrieval side is working as well as current technology allows.

Answer-aware retrieval: Developing retrievers that explicitly optimize for answer-bearing evidence, beyond topical similarity, may reduce distracting and contradictory context, and better align retrieval with the downstream prediction task.

Contradiction detection and filtering: Integrating explicit contradiction detection prior to generation could prevent the most harmful failure type, in which retrieved evidence directly conflicts with the correct answer, particularly for safety-critical topics such as drug contradictions.

Cross-domain validation: Extending our analysis to other specialized domains (e.g., legal, financial, or scientific RAG) would test whether the identified failure modes and mitigation techniques generalize beyond medical question answering.

Knowledge distillation from robust teachers: Given that our robust $k = 5$ configuration achieves substantially improved performance, a natural next step is to distill its behavior into a smaller student model, yielding a more efficient yet robust medical QA system.

Automated robustness-oriented prompt design: Finally, exploring automated strategies for constructing robustness-focused prompts—for example, using search over prompt templates or validation-guided refinement—could make generator-side mitigation more scalable and less dependent on manual engineering.

VI. Ethical Constraints of Medical RAG Systems

The system's accuracy of 59.6% which means the system gives wrong answer for 4 out of 10 questions which is questionable for safe clinical use. Patient treatment is seriously jeopardized by confident misinformation. Additionally, safety testing and validated trials must come before deployment. RAG systems occasionally rely on out-of-date and inadequate sources. Automation bias lowers independent clinical judgment and raises the possibility that users may over-trust AI advice. Human-in-the-loop verification must be required for any deployment that is safety-critical.

Conclusion

We successfully investigated the failure modes of RAG pipelines in specialized medical question answering. We demonstrated that a standard RAG implementation with naive prompting can be detrimental to performance, with accuracy collapsing from 68.8% (golden context) to as low as 17.6% as more retrieved documents are added. Through systematic analysis of 165 RAG-induced failures, we identified that this degradation is primarily caused by retrievers providing "Distracting" (41.8%) and "Contradictory" (37.6%) context. Most importantly, we demonstrated that this failure is not fundamental to RAG systems but is a consequence of insufficient generator robustness. Through advanced few-shot prompt engineering, we not only prevented performance collapse but reversed the trend, enabling the model to benefit from additional context. Our robust k=5 configuration achieved 59.6% accuracy, representing a 186% improvement over the naive approach (20.8%) and approaching the golden context baseline (68.8%).

We draw the conclusion that the generating component in high-stakes domains needs to be specifically designed to be resilient to the retrieval component's unavoidable flaws. This discovery causes the optimization focus to change from only improving retrieval quality to also improving generator robustness, allowing prompt engineering to unlock considerable performance advantages.

References

[1] Singhal, K., Azizi, S., Tu, T., *et al.* (2023). Large language models encode clinical knowledge. *Nature*, 620, 172–180. doi: 10.1038/s41586-023-06291-2

[2] Thirunavukarasu, A.J., Ting, D.S.J., Elangovan, K. *et al.*, Large language models in medicine. *Nature Medicine* 29, 1930–1940 (2023). doi: 10.1038/s41591-023-02448-8

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

[4] Ram, O., Levine, Y., Dalmedigos, I., Muhlgay, D., Shashua, A., Leyton-Brown, K., & Shoham, Y. (2023).*In-context retrieval-augmented language models.Transactions of the Association for Computational Linguistics*, 11, 1316–1331.

[5] Jin, Q. *et al.*, "PubMedQA: A dataset for biomedical research question answering," in *Proc. EMNLP-IJCNLP*, 2019, pp. 2567–2577.

[6] Agrawal, M., Hegselmann, *et al.*, "Large language models are few-shot clinical information extractors," in *Proc. EMNLP*, 2022,arXiv:2205.12689.

[7] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. EMNLP-IJCNLP*, 2019, pp. 3982–3992,arXiv:1908.10084.

[8] J. Johnson, M. Douze, and H. Jégou,"Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, 7(3), 535–547, 2021, doi: 10.1109/TBDATA.2019.2921572.

[9] H. Chur *et al.*, "MMed-Llama-3-8B: A specialized medical language model," Hugging Face, 2024.

[10] T. B. Brown *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 1877–1901, doi: 10.48550/arXiv.2005.14165.

[11] J. Wei *et al.*, "Finetuned language models are zero-shot learners," in *Proc. ICLR*, 2021, doi: 10.48550/arXiv.2109.01652.

[12] O. Yoran *et al.*, "Making retrieval-augmented language models robust to irrelevant context," *arXiv preprint* arXiv:2310.01558, 2023.

[13] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open-domain question answering," in *Proc. EACL*, 2021, pp. 874–880, doi: 10.18653/v1/2021.eacl-main.74.

[14] W. Shi *et al.*, "REPLUG: Retrieval-augmented black-box language models," *arXiv preprint* arXiv:2301.12652, 2023.

[15] K. Singhal *et al.*, "Towards expert-level medical question answering with large language models," *arXiv preprint* arXiv:2305.09617, 2023, doi: 10.1038/s41591-024-03423-7.

[16] H. Xiong *et al.*, "DoctorGLM: Fine-tuning your Chinese doctor is not a herculean task," *arXiv preprint* arXiv: 2304.01097, 2024, doi = 10.48550/arXiv.2304.01097.

[17] R. Nogueira and K. Cho, "Passage re-ranking with BERT," *arXiv preprint* arXiv:1901.04085, 2019.

[18] M. Glass *et al.*, "Robust retrieval augmented generation for zero-shot slot filling," in *Proc. EMNLP*, 2021, pp. 1939–1949, doi = 10.18653/v1/2021.emnlp-main.148.

[19] X. Ma *et al.*, "Query rewriting for retrieval-augmented large language models," in *Proc. EMNLP*, 2023, pp. 5303–5315, doi: 10.48550/arXiv.2305.14283.

[20] World Health Organization, *Ethics and Governance of Artificial Intelligence for Health*. Geneva, Switzerland: WHO Press, 2021.

[21] U.S. Food and Drug Administration, "Clinical Decision Support Software: Guidance for Industry and FDA Staff," 2022. docket: FDA-2017-D-6569.

[22] C. Benjamens, P. Dhunnoo, and B. Mesko, "The state of artificial intelligence-based FDA-approved medical devices and algorithms: An online database," *NPJ Digit. Med.*, vol. 3, no. 1, pp. 1–8, 2020, doi: 10.1038/s41746-020-00324-0.

[23] O. Agafonov *et al.*, "Editorial: Trustworthy AI for healthcare," *Front. Digit. Health*, vol. 6, Art. no. 1427233, 2024, doi: 10.3389/fdgth.2024.1427233.

[24] L. Weidinger *et al.*, "Taxonomy of risks posed by language models," in *Proc. ACM Conf. Fairness, Accountability, and Transparency (FAccT)*, pp. 214–229, 2022, doi: 10.1145/3531146.3533088.

[25] J. Sohn *et al.*, "Rationale-guided retrieval augmented generation for medical question answering," *arXiv preprint arXiv:2411.00300*, 2024, doi: 10.48550/arXiv.2411.00300.

Appendix

This appendix provides the final robust prompt used in Phase 3 experiments. The prompt enforces selective use of medically relevant evidence, rejection of unsupported or contradictory statements, and strict format compliance with the PubMedQA answer schema: {yes, no, maybe}.

A. Core Instruction

> You will be given a medical question and several retrieved evidence passages. Some evidence may be irrelevant, ambiguous, or contradictory. Use only the evidence that directly supports the correct answer. Ignore distracting information. If the correct answer is unclear or conflicting, respond "maybe." Your response must be exactly one token: yes, no, or maybe.

B. Few-Shot Examples (Abbreviated)

Example 1 (Supported Evidence — Answer = yes):
Context: Evidence explicitly reports association between Treatment A and improved Outcome B.
Question: Does Treatment A improve Outcome B?
Answer: yes

Example 2 (Ambiguous Evidence — Answer = maybe):
Context: Findings are mixed or inconclusive across multiple studies.
Question: Is Biomarker X related to Disease Y?
Answer: maybe

Example 3 (Contradictory Evidence — Answer = no):
Context: Evidence consistently reports no significant association between Variable C and Condition D.
Question: Does Variable C influence Condition D?
Answer: no

This prompt was selected after iterative refinement informed by our failure mode analysis. It serves as the basis for mitigation results reported in Section V.

We provide four representative error cases demonstrating the primary failure modes identified in our qualitative analysis.

C. Case Study A: Contradictory Context

Question: The HELLP syndrome—evidence of a possible systemic inflammatory response?
Ground Truth: yes
Prediction (Naive RAG): no

Retrieved Context (excerpt): A report highlighting "no significant association" across inflammatory markers and subgroups.

Failure Mechanism: The retrieved content directly contradicted the correct answer. Misleading negative phrasing caused the model to align with incorrect evidence. This reflects the 37.6% of failures attributed to contradictory context.

D. Case Study B: Distracting/Ambiguous Context

Question: Do instrumental activities of daily living predict dementia at 1–2 years?
Ground Truth: yes
Prediction (Naive RAG): maybe

Retrieved Context (excerpt): A study comparing IADL performance in MCI vs. normal controls, focused on current status rather than prospective prediction.

Failure Mechanism: The context was topically relevant but not answer-bearing. Lack of temporal clarity caused the model to hedge. This aligns with the 41.8% of failures due to distracting/ambiguous context.

E. Case Study C: Irrelevant Context

Question: Do cytokine concentrations in pancreatic juice predict pancreatic disease?
Ground Truth: yes
Prediction (Naive RAG): no

Retrieved Context (excerpt): Analysis of serum cytokeratin-18 in pancreatitis severity — a different biomarker and fluid source.

Failure Mechanism: Superficial keyword overlap misled the retriever. The generator followed confidently-stated irrelevant evidence, contributing to the 13.9% of failures in this category.

F. Case Study D: Context-Induced Unparsable Response

Question: Gluten tolerance in adult celiac disease—does it occur?
Ground Truth: maybe
Prediction (Naive RAG): Unparsable (long-form explanation)

Retrieved Context (excerpt): Mixed reporting of heterogeneous patient responses and diagnostic uncertainty.

Failure Mechanism: Conflicting information overwhelmed the model, leading to abandonment of the yes/no/maybe schema. This accounts for the 7.3% of failures classified as unparsable responses.

These case studies demonstrate that RAG-induced failures arise from systematic weaknesses in the retrieval component that propagate through the generation stage, reinforcing the need for generator robustness.