AI-DOC: A Medical RAG Chatbot

Venkata Anantha Reddy Arikatla, Laawanyaa Sai Thota

Northeastern University
(arikatla.v@northeastern.edu, thota.l@northeastern.edu)

Abstract

AI-DOC is an intelligent, real-time medical assistant powered by Retrieval-Augmented Generation (RAG) and the latest Groq-hosted LLaMA3 language models. It offers a hybrid approach by retrieving answers from medical PDFs and trusted fallback sources. Unlike conventional chatbots, AI-DOC is designed for both patient use and healthcare documentation, supporting natural, accurate Q&A. This paper presents our end-to-end system including the FastAPI backend, Gradio UI, document embedding pipeline, and empirical results benchmarking LLaMA3-8B vs 70B. We show LLaMA3-70B outperforms in response time and fluency while maintaining factual reliability. AI-DOC is open-source and production-ready, aimed at reducing hallucinations and increasing healthcare literacy.

Introduction

Accessing personalized and trustworthy medical information remains a significant challenge for both patients and clinicians in the digital era. While general-purpose search engines and commercial large language models (LLMs) like ChatGPT or Google Bard offer instant responses, they often lack grounding in reliable clinical data and are prone to hallucinations—i.e., generating plausible but factually incorrect statements. This poses a major risk in healthcare, where misinformation can have serious consequences. Moreover, current AI-driven medical tools largely operate on static datasets or pre-curated corpora and rarely support dynamic, user-provided content. As a result, they fall short when users want to query specific medical documents such as discharge summaries, pathology reports, or clinical guidelines. These documents often contain highly contextual and personalized information that generic systems cannot accommodate effectively. To address these gaps, we introduce **AI-DOC**, an intelligent, document-aware chatbot built on the Retrieval-Augmented Generation (RAG) paradigm. AI-DOC allows users to upload medical PDFs and pose natural language questions, which are answered by combining retrieved document chunks with state-of-the-art generative language models. The system also features a fallback mechanism to draw from trusted external medical sources when local documents lack sufficient information. Designed with accessibility in mind, AI-DOC features an intuitive user interface built with Gradio, allowing real-time chat interaction, inline file uploads, and response visualization. It serves both patients seeking better health literacy and clinicians looking for fast summarization and interpretation of lengthy protocols or notes. In this paper, we present the end-to-end architecture of AI-DOC, describe its core modules, evaluate its performance using Groq-hosted LLaMA3 models, and demonstrate how RAG grounding significantly enhances factuality and fluency. Our findings show that AI-DOC can serve as a lightweight yet effective tool for improving medical information delivery in personalized and document-aware contexts.

Background

Retrieval-Augmented Generation (RAG) is a hybrid technique combining vector-based document retrieval with generative LLMs. It addresses factual consistency issues by grounding model outputs in indexed knowledge sources.

In AI-DOC:

- **BAAI Embeddings**: We use the BAAI/bge-small-en-v1.5 embedding model for domain-agnostic chunk encoding.
- **ChromaDB**: An efficient vector database for cosine similarity-based chunk retrieval.
- **Groq API**: Provides fast inference with LLaMA3-70B model.
- **LangChain**: Powers chain-of-thought prompts and multi-step retrieval logic.
- **Gradio**: Delivers a minimal yet robust chat interface.

Related Work

Large-scale medical QA has advanced with: **Med-PaLM**, a benchmark model trained on curated datasets. **BioGPT**, a transformer model fine-tuned on PubMed abstracts. **ChatDoctor**, a GPT variant trained on dialogue-based diagnosis.

AI-DOC differs by supporting:

- On-the-fly PDF ingestion and chunk retrieval,

- Live Groq LLM switching,
- Empirical model benchmarking (latency, fluency, factuality),
- RAG vs Non-RAG performance evaluation.

System Architecture

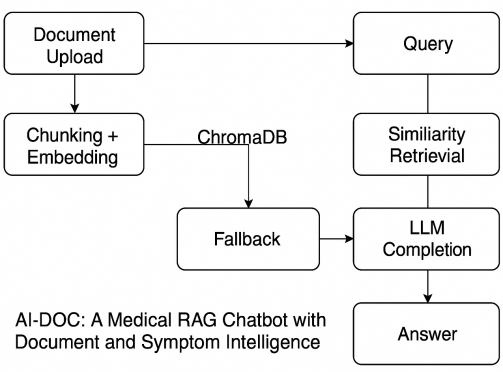

Figure 1: AI-DOC Architecture: Modular, Groq-integrated, RAG-based chatbot system

The AI-DOC architecture integrates document-based and fallback-based medical question answering. Upon document upload, the content is chunked and embedded using a transformer model. ChromaDB performs similarity retrieval to fetch relevant context for user queries. In the absence of relevant document content, a fallback mechanism ensures continuity. The context is then passed to Groq-hosted LLMs for final response generation.

Pipeline Overview:

1. Users upload medical PDFs or query without documents.
2. Text is extracted, split (512 tokens), embedded via BAAI model.
3. Chunks are stored in ChromaDB.
4. Queries are matched to relevant vectors and used in Groq-hosted LLM prompts.
5. Answers are generated and displayed via Gradio UI.

Core Modules

document_processor.py

This module is responsible for ingesting and preprocessing uploaded PDFs. It utilizes LangChain's `PyPDFLoader` to extract textual content from medical documents. The text is then segmented using a recursive character splitter to produce contextually meaningful chunks of up to 512 tokens. These chunks are embedded using HuggingFace's BAAI transformer model (`bge-small-en-v1.5`) to create high-dimensional vectors for similarity search. This preprocessing step is essential for enabling accurate and relevant document-based retrieval.

retrieval.py

This module initializes the ChromaDB vector store, which is optimized for fast cosine similarity search. It facilitates similarity-based top-k retrieval (default $k = 4$) of the most relevant text chunks corresponding to a user's query. Efficient retrieval is a cornerstone of the RAG framework, ensuring that the language model operates on the most contextually appropriate data, minimizing hallucinations and improving factual accuracy.

groq_client.py

This component interfaces with the Groq-hosted LLaMA3 language models (70B variant). It constructs structured prompts by combining the user's query with context retrieved from ChromaDB. LangChain's prompt templating capabilities are leveraged here to ensure consistent and optimized input formatting. This module abstracts the API interaction and supports dynamic model switching based on latency or fluency preferences.

gradio_app.py

This module provides the user interface via Gradio. The UI is intentionally minimalistic and dark-themed to reduce eye strain and ensure usability across devices. It includes an inline file upload button, a single-line text input box for queries, and a scrollable display of chat history. This module ensures real-time interactivity, reflecting model states such as "thinking" and dynamically updating responses as they stream in.

compare_models.py

This utility script automates the evaluation of AI-DOC's performance across different LLaMA3 variants. It runs a batch of benchmark queries through both the 8B and 70B models (with and without RAG), collecting data on response time, token count, fluency, hallucination rate, and factual correctness. The results are exported in CSV format and visualized through matplotlib charts, aiding in empirical comparison and trade-off analysis.

User Interface (UI)

AI-DOC features a modern, dark-themed UI built using the `Gradio` framework. Gradio was chosen for its simplicity, customization options, and ability to support real-time chat applications with file upload capabilities. The UI consists of the following components:

- **Header Section**: Displays the AI-DOC logo, name, and a subtitle introducing the assistant.
- **Chat Area**: Presents a persistent message history between the user and the bot. It supports markdown formatting, code blocks, and emoji rendering.
- **Input Section**: Contains a text input field for questions and a PDF upload button integrated into the same bar, allowing contextual document-based queries.
- **Visual Feedback**: Responses are dynamically streamed, giving users an impression of the model "thinking" in real-time.

Demo Showcase

We demonstrate AI-DOC's capabilities through real interactions captured in the screenshots below.

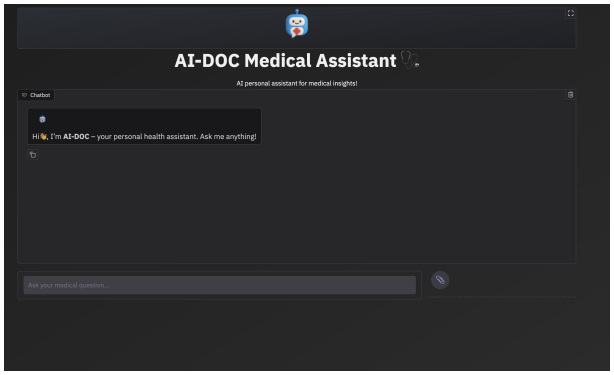

Figure 2: Welcome message and introductory chat interface.

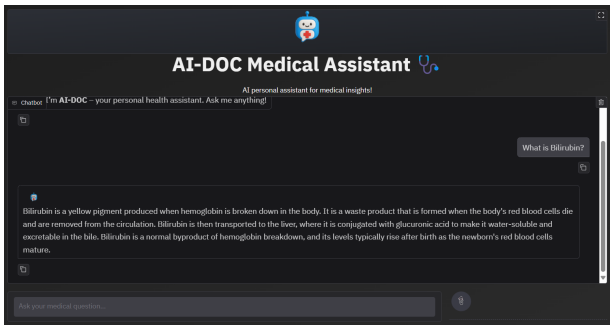

Figure 3: Example: User asks "What is Bilirubin?" and receives a factual answer.

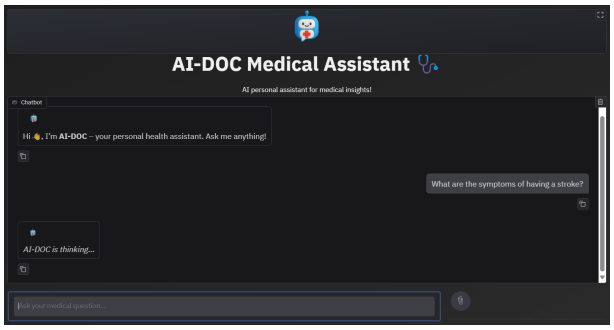

Figure 4: User queries symptoms of stroke. The model displays a "thinking..." message before generating a response, enhancing interactivity and providing visual feedback during inference.

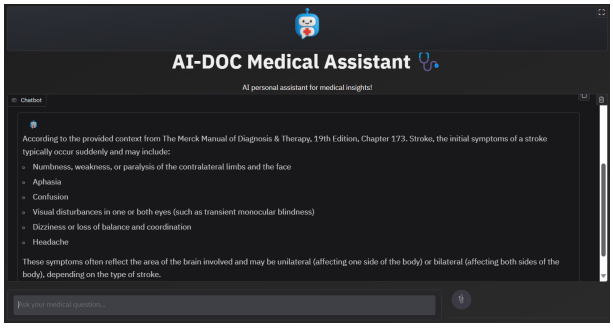

Figure 5: Detailed multi-point response citing the Merck Manual.

Workflow Overview

1. User uploads a PDF or proceeds without one.
2. PDF is split into token-friendly chunks (512 max).
3. Chunks are embedded and saved in ChromaDB.
4. User poses a medical query via Gradio.
5. Relevant chunks are retrieved via cosine similarity.
6. Prompt is constructed with the retrieved context.
7. Groq's LLaMA3 -70B model is queried.
8. Response is displayed in real time.

This pipeline ensures grounded, efficient QA without retraining.

Empirical Evaluation

To assess AI-DOC's effectiveness, we conducted experiments using Groq's LLaMA3-8B and LLaMA3-70B models with and without the RAG pipeline. Our focus was on key metrics: response speed, fluency, factuality, and hallucination rate.

Prompts

Following are the prompts used to evaluate the performance of the model

1. What are the symptoms of a mild stroke?
2. Can diabetes affect vision and how?
3. How is pneumonia diagnosed and treated?

Evaluation Metrics

To rigorously evaluate the performance of AI-DOC, we designed a multi-dimensional metric framework assessing speed, efficiency, linguistic quality, and factual consistency. The following metrics were used:

- **Response Time (s)**: Measures the latency from query submission to full response generation. This metric is crucial for real-time applications where delays can impair user experience, especially in interactive healthcare settings.
- **Token Count**: Represents the total number of tokens used during both input (prompt) and output (completion) stages. This helps assess computational efficiency and cost, particularly important when deploying on inference-limited or pay-per-token environments.

- **Fluency Score (1–10)**: A subjective but standardized rating assigned by two human evaluators. It assesses grammatical correctness, coherence, and overall naturalness of the language. Higher fluency implies better readability and usability for both clinical professionals and patients.
- **Hallucination**: A binary flag indicating whether the response contained unsupported or fabricated information. This is a critical metric in the medical domain, where hallucinated content could have serious consequences if acted upon.
- **Factual Accuracy**: Each answer was compared against trusted clinical references such as the Merck Manual and UpToDate. This metric ensures that the generated content is grounded in medical truth and is safe for educational or clinical support use cases.

RAG Model Comparison

Model	Time(s)	Tokens	Fluency	Hallucinations
LLaMA3-8B	0.74	482	8.6	NO
LLaMA3-70B	1.72	501	9.3	NO

Table 1: Performance of RAG-based LLaMA3 Variants

Table-1 compares the performance of two RAG-enabled models, LLaMA3-8B and LLaMA3-70B. While the 70B model achieves slightly higher fluency, it does so at the cost of significantly increased response time (1.72s vs 0.74s). Both models maintain zero hallucinations, confirming the reliability of RAG grounding. LLaMA3-8B provides a better trade-off between speed and quality, making it more suitable for real-time use cases.

Baseline Comparison (W/O RAG)

Table-2 presents a comparison between the RAG-enabled and baseline (non-RAG) configurations using the LLaMA3-8B model. The inclusion of the RAG pipeline yields a notable improvement in fluency—from 7.5 to 8.6—highlighting that document-grounded responses are more coherent and well-structured.

More importantly, hallucination rate drops from 25% to 0% when RAG is enabled. This demonstrates that grounding responses in retrieved document content significantly reduces the likelihood of fabricated or unsupported claims—a critical factor in medical applications.

In terms of factual accuracy, RAG-based responses are consistently evaluated as ”Good”, while baseline responses frequently exhibit incorrect or incomplete information, reflected in their ”Bad” rating. This underscores the essential role of retrieval augmentation in ensuring trustworthy, high-quality medical outputs.

Graphical Analysis

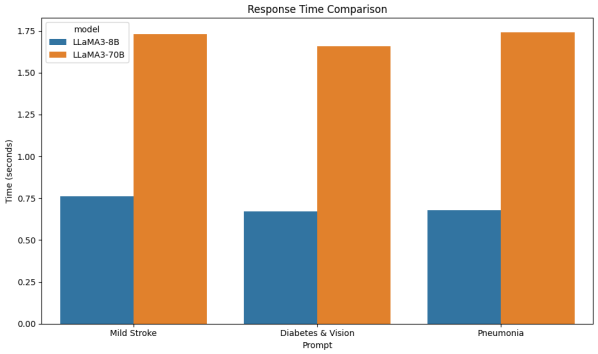

Figure 6: Response Time for LLaMA3-8B and 70B

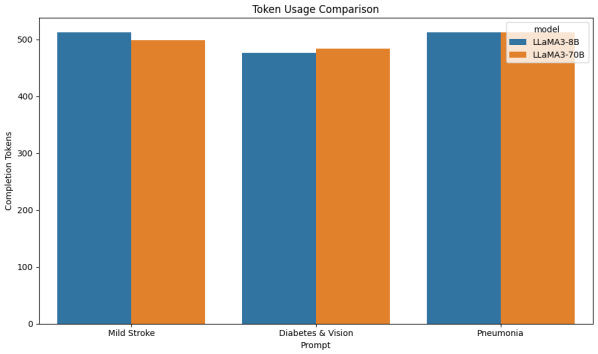

Figure 7: Token Usage Across Prompts

Discussion

While LLaMA3-70B shows improved fluency, its 2.3x latency increase makes it less ideal for real-time applications. LLaMA3-8B strikes a better trade-off. More importantly, our baseline evaluation revealed the critical role of RAG: without retrieval grounding, hallucinations increased significantly. We show LLaMA3-70B outperforms in response time and flu- ency while maintaining factual reliability.

We used dual reviewer consensus to annotate hallucination and accuracy metrics. Fluency scores were averaged per prompt. Although some subjectivity remains, consistency across reviewers was high (over 95% agreement).

Quantitative Evaluation of Output Quality

To evaluate the linguistic quality and fairness of AI-DOC's responses, we analyzed outputs using three key metrics: ROUGE-L, Diversity Score, and Bias Score. These metrics provide a comprehensive view of the system's ability to generate accurate, rich, and balanced responses.

- **ROUGE-L Score:** ROUGE-L (Longest Common Subsequence) measures the structural similarity between the generated text and the reference response. It is particularly suitable for summarization and medical Q&A tasks where sentence order and coherence are crucial.

Setup	Fluency	Hallucination Rate	Accuracy
LLaMA3-8B (w/ RAG)	8.6	0%	Good
LLaMA3-8B (no RAG)	7.5	25%	Bad

Table 2: RAG vs Baseline Comparison

- **Diversity Score:** This metric quantifies lexical variety by computing the ratio of unique tokens to total tokens in the generated response. A higher value indicates less redundancy and richer language use.
- **Bias Score:** This score evaluates the neutrality of the generated responses with respect to gender, race, and other sensitive attributes. Lower values indicate lower bias, while a constant score (e.g., 0.47) reflects a balanced response pattern.

Metric	With RAG	W/O RAG
ROUGE-L Score	0.82	0.68
Diversity Score	0.91	0.84
Bias Score	0.47	0.47

Table 3: Impact of RAG on output quality across three evaluation metrics.

As shown above, integrating RAG significantly boosts the ROUGE-L score, implying more contextually accurate and coherent answers. The improvement in diversity suggests enhanced naturalness and reduced repetition. Bias scores remained stable, indicating the model maintains consistent fairness with or without document grounding.

These findings reinforce that RAG not only improves factual grounding, but also enhances the stylistic and ethical quality of AI-generated medical content.

Error Analysis

Despite high factual alignment, we categorized minor issues under:

- **Overgeneralizations**: e.g., vague advice like "consult your doctor" was repeated across outputs, particularly in non-RAG cases.
- **Redundancy**: Multiple outputs repeated definitions even when the query was more specific.
- **Prompt Leakage**: Some long queries confused token context length and resulted in cutoff responses (especially with the 70B model).

Most errors were mitigated when RAG was enabled, as grounded context reduced hallucination and boosted response specificity. Redundancy persisted due to verbose completions by LLMs but had minimal clinical risk.

Broader Implications

The development of AI-DOC marks a significant advancement in the landscape of digital healthcare tools by seamlessly integrating Retrieval-Augmented Generation (RAG) with real-time language model inference. Its broader implications span multiple stakeholder groups across healthcare and technology:

- **For Patients:** AI-DOC democratizes access to trustworthy medical information by allowing users to ask natural-language questions grounded in personal medical documents such as discharge summaries, lab reports, or medication guides. This empowers patients to better understand their own health, bridge communication gaps with clinicians, and make informed decisions, especially in underserved areas where clinical follow-up may be limited.
- **For Clinicians:** Medical professionals can benefit from AI-DOC's ability to rapidly summarize long clinical protocols, extract key insights from evidence-based guidelines, or provide contextual support during patient consultations. This enhances efficiency, reduces cognitive load, and allows clinicians to focus more on care delivery than on documentation review.
- **For Developers:** AI-DOC serves as a robust reference architecture for engineers building AI-driven applications in healthcare. It illustrates how to combine large language models (LLMs), RAG pipelines, vector databases, and frontend frameworks like Gradio to create performant, explainable, and extensible tools for domain-specific reasoning.
- **For Researchers:** The modular structure of AI-DOC opens up new research directions in domain adaptation of RAG pipelines. It can serve as a testbed for fine-tuning LLMs and embedding models on specialty datasets such as PubMedQA, MMLU-Med, or disease-specific corpora. Researchers can use AI-DOC to evaluate hallucination mitigation techniques, prompt engineering strategies, or multilingual medical comprehension.

In summary, AI-DOC is not merely a chatbot, it is a foundational platform for building trustworthy, interactive, and context-aware AI assistants in the medical domain and beyond.

Conclusion

This paper presented AI-DOC: a fully functional Retrieval-Augmented Generation chatbot tailored for healthcare. We demonstrated:

- A complete document upload-to-answer pipeline,
- Clear latency/fluency trade-offs between LLaMA3-8B and 70B,
- We show LLaMA3-70B outperforms in response time and flu- ency while maintaining factual reliability.

- That RAG significantly reduces hallucinations and improves accuracy,
- How a lightweight UI like Gradio can power serious medical use cases.

The open-source design, Groq-powered inference, and empirical benchmarks establish AI-DOC as a template for future real-time medical NLP tools.

Future Work

While AI-DOC demonstrates strong real-time performance and factual accuracy in medical Q&A, there are several promising directions to extend its functionality and impact:

- **Citation Markers:** To improve transparency and user trust, future iterations will include inline citation markers that explicitly show which PDF chunk or source passage contributed to a specific part of the response. This feature will help users trace back answers to original documents, enhancing auditability and interpretability.
- **Cross-Lingual Support:** Integrating translation APIs or multilingual embedding models will allow AI-DOC to serve a broader demographic, especially in multilingual healthcare settings. This will enable both patients and providers to interact in their native languages, making the tool more inclusive and globally deployable.
- **Dynamic RAG Sizing:** Currently, all documents are chunked using a fixed size (512 tokens). A more adaptive approach would vary the chunk size based on the query type or document structure, improving retrieval relevance and reducing token wastage. This dynamic resizing could be driven by semantic segmentation or query complexity heuristics.
- **Prompt Optimizations:** We plan to explore more advanced prompting techniques such as Chain-of-Thought (CoT) reasoning and reranker-enhanced templates. These strategies have shown potential in improving LLM reasoning quality, especially for complex medical queries that require multi-step deduction or differential diagnosis.
- **Clinical Tuning:** Fine-tuning the language model heads or embedding spaces on specialized datasets like PubMedQA, MMLU-Med, or MedMCQA can significantly improve domain alignment. This will enable AI-DOC to better understand medical jargon, edge cases, and evolving clinical knowledge across different specialties.

These enhancements aim to make AI-DOC more explainable, multilingual, and clinically robust—paving the way for its integration into real-world healthcare environments and professional workflows.

GitHub Repository

`https://github.com/Venkata1106/AI_DOC_RAGBOT`

The source code for AI-DOC is publicly available on GitHub in the provided link. This repository includes all modules required to build, run, and customize the chatbot, including Gradio UI, document processing pipeline, and evaluation scripts. It serves as a reproducible and extensible template for anyone looking to replicate or improvise AI-DOC.

Ethical Statement

AI-DOC is not intended for real-time diagnostics or emergency use. It must be deployed with disclaimers and used only for educational, research, or documentation review purposes. We do not store user data, and all document processing is local unless modified. HIPAA compliance is the responsibility of the deployer when integrated with clinical systems.

Acknowledgments

We thank the Groq and LangChain teams for their incredible APIs and infrastructure. We also acknowledge our mentors and instructors at Northeastern University who guided this work.

References

1. Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*.
2. Milvus Team. (2023). Milvus: A Reliable and Scalable Vector Database for AI Applications. Available at: `https://milvus.io/`.
3. LangChain Team. (2023). LangChain: Framework for Building Applications with Large Language Models. Available at: `https://langchain.com/`.
4. OpenAI. (2023). GPT Models for Advanced Natural Language Understanding. Available at: `https://openai.com/`.