Optimized Retrieval-Augmented Generation Framework for Enhanced Medical Query Processing

Aarthi M
Department of Computer Science
(MTECH CSE)
Vellore Institute of Technology
Vellore,Tamil Nadu -632014
aarthimanoharan2003@gmail.com

Riddhi Gindodiya
Department of Computer Science
(MTECH CSE)
Vellore Institute of Technology
Vellore,Tamil Nadu -632014
riddhigindodiya06@gmail.com

Anmol Singh
Department of Computer Science
(MTECH CSE)
Vellore Institute of Technology
Vellore,Tamil Nadu -632014
mranmolsingh101@gmail.com

***Abstract*—Large language models (LLMs) have been a game-changer in a number of fields in recent years, including healthcare and medical education. This work offers a case study on the real-world implementation of retrieval-augmented models for improving healthcare education in low- and middle-income nations that are based on generation (RAG). The need for easily available and locally relevant medical information to support community health workers in providing high-quality maternity care led to the development of the SMARThealth GPT model, which is the subject of this research. We outline the whole RAG pipeline development process, which includes parameter selection and optimization, knowledge embedding retrieval, response production, and the establishment of a knowledge base of Indian pregnancy-related rules. This case study demonstrates how LLMs may improve guideline-based health education and develop the ability of frontline healthcare workers. It also provides ideas for comparable applications in environments with restricted resources. It is a resource for machine learning researchers, teachers, medical experts, and legislators who want to use LLMs to significantly enhance education.**

Keywords—Machine Learning, Large language Models, Retrieval Augmented Generation, Natural Language processing, Medical Assisstent.

I .INTRODUCTION

Large Language Models (LLMs) are the solution for majority of the text-related tasks or LLMs, are the standard approach. Their factual accuracy1, a drawback of their generative nature, is still a serious worry, nevertheless. LLMs are made to produce believable text based on learnt patterns rather than to acquire exact facts [1]. Contextualizing LLMs by the use of pertinent input tokens to affect their output is a common method of improving their factuality. This includes more complex Retrieval Augmented Generation (RAG) methods as well as more straightforward prompting strategies like "Let's think step by step." Context retrieval system integration may, in fact, greatly improve LLM performance and dependability[1].

Recently, With the growing availability of pre-trained large language models (LLMs), including Open AI's GPT, Lama, and PaLM, the field of natural language processing (NLP) has recently witnessed amazing advancements. These models have been used in a variety of sectors and are becoming more and more working in healthcare and medical education. Two effective techniques for adapting pre-trained LLMs to particular applications are retrieval-augmented generation (RAG) and fine-tuning. In a "close-book" scenario, fine-tuning adjusts the model's weight according to a task-specific dataset, depending only on extra input-output pairs of training data for learning. On the other hand, RAG does not require labeled training data and functions in a "open-book" environment.

A. What is RAG

The implementation of goal-oriented large language models (LLMs) in conjunction with various LLM-oriented frameworks is expanding the range of AI applications and improving LLMs' ability to perform complicated tasks. Modern LLMs are quite capable, ranging from chatbots that can generate programming code to responding to inquiries on legal papers with latent provenance. But this enhanced potential also brings with it new complications. Despite their strength at traditional text-based activities, emerging LLMs require outside assistance to keep up with changing knowledge [2].

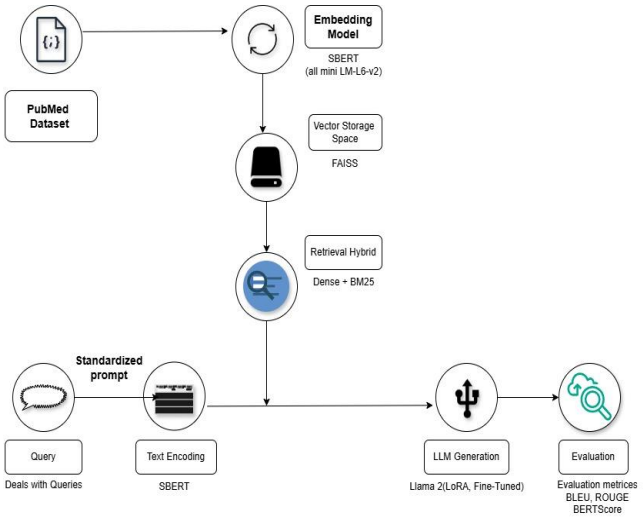

Fig. 1. RAG Model

Non-parametric retrieval-based approaches, like as retrieval-augmented generation (RAG), are becoming essential to the most recent LLM applications in order to overcome this difficulty, particularly for domain-specific tasks.

The development of AI-stack applications emphasizes how important it is to improve RAG techniques in order to keep LLMs' knowledge bases up to date. When using semantic similarity search to find the most pertinent passages, or top-K vectors, retrieval-based applications require optimization. There are dependencies on time and token constraints when querying multi-document vectors and adding pertinent context to LLMs. The "bi-encoder" retrieval models make use of state-of-the-art approximation nearest-neighbor techniques[4].

B. Related work

Numerous studies have been conducted in an effort to address the problem of LLM factuality. Using LLMs' innate In-Context Learning (ICL) capabilities was the main focus of early attempts to enhance it, enabling individuals to adjust to new duties without particular training and with few examples. This opened the door for the creation of complex prompting strategies intended to elicit more precise and thoughtful answers. LLMs do better on challenging problems when guided through intermediate reasoning processes by Chain of Thought (CoT) prompts. Self-Consistency (SC), on the other hand, takes use of the stochastic character of LLMs by generating and contrasting several results for the same input before generating a single, cohesive response [3]. The Self-Consistency Chain of Thought (SC-CoT) combines both.
Researchers used to prompt strategies to integrate external knowledge after realizing the limitations of relying just on internal knowledge, which eventually gave rise to retrieval RAG stands for Augmented Generation. By biasing replies with real data, RAG systems greatly improve LLM performance by retrieving and integrating pertinent information from external knowledge sets. Medprompt a context retrieval system created for medical MCQA that produces state-of-the-art answers with GPT-4, proposes a combination of few-shot, CoT, and SC, which are frequently utilized in the healthcare area to increase factuality. Although Medprompt has been modified for open-source models, a comprehensive analysis of the best way to set up its constituent parts (such as DBs and embeddings) is still a work in progress[4].

II .LITERATURE SURVEY

A. Existing Approaches for Large-Scale Knowledge Bases

Many techniques and tactics are now used to manage large-scale knowledge bases, each of which is intended to address specific challenges associated with processing massive volumes of textual material. These techniques may be broadly categorized into many key strategies.

Searching and indexing algorithms: Conventional information retrieval methods rely on indexing strategies such as inverted indexes and search algorithms such as TF-IDF (Term Frequency-Inverse-Document-Frequency) to efficiently locate relevant documents inside large knowledge repositories. Many information retrieval systems are built on these processes, which allow for the prompt and precise retrieval of information in response to user queries[3].

Computing framework: Computing frameworks such as Apache Hadoop and Spark have made it easier to manage large-scale knowledge base processing and analysis. These frameworks enable the parallel processing of data over several nodes, enabling the efficient and scalable computation of complex tasks such as indexing, querying, and analysis.
NLP and ML Techniques: To glean insights from vast volumes of text data, deep learning architectures like transformers, in addition to other cutting-edge machine learning and NLP models, are increasingly being employed. Models that excel at tasks like text classification, summarization, and question answering, such as B-E-R-T (Bidirectional-Encoder-Representations from Transformer) and GPT (Generative Pre-trained Transformer), can manage large knowledge bases. Knowledge graphs are structured representations of knowledge that hold entities, relationships, and characteristics using a graph-based structure. By organizing data into connected nodes and edges, knowledge graphs make it easier to efficiently navigate and retrieve relevant information from large sources. When knowledge graphs are filled and improved with strategies like these, they become more beneficial for knowledge retrieval tasks.
Mixed techniques: Several contemporary techniques use elements of the aforementioned strategies in order to optimize the advantages of different approaches. For example, hybrid systems can mix machine learning models with traditional indexing methods or leverage distributed computing frameworks to increase the scalability of knowledge retrieval and analysis processes[3].

B. LLMs in Medical Domains

Large Language Models (LLMs) have emerged as powerful tools in the medical domain, transforming how healthcare professionals, researchers, and patients access and interpret complex medical information. These models, trained on massive datasets, including scientific literature, clinical notes, and public health data, can understand, generate, and summarize medical content with remarkable accuracy. In clinical decision support, LLMs assist physicians by providing evidence-based answers to diagnostic queries, suggesting treatment options, and analyzing patient data for potential risks. They are also invaluable in biomedical research, helping researchers navigate vast amounts of literature by generating insights and summaries from multiple sources, including databases like PubMed[2].
LLMs contribute to patient engagement by simplifying medical jargon into understandable language, empowering patients to make informed decisions about their health. Despite their immense potential, LLMs face challenges, such as ensuring data privacy, managing biases in training data, and maintaining up-to-date medical knowledge [3]. Furthermore, the need for regulatory compliance and validation of AI-generated medical advice underscores the importance of human oversight. As LLMs continue to evolve, their integration into the medical domain holds great promise for advancing healthcare delivery, research efficiency, and patient outcomes.

C. RAG methods

Retrieval-Augmented Generation (RAG) is a hybrid approach in natural language processing that combines information retrieval with language generation to produce more accurate and contextually relevant responses. Unlike traditional language models that rely solely on pre-trained knowledge, RAG dynamically retrieves external information from large datasets or document repositories to augment the generation process. This makes it particularly suitable for tasks requiring factual accuracy and domain-specific knowledge, such as biomedical literature search, customer support, and legal document analysis[1].

1. Stuff Method

The stuff method directly concatenates all the retrieved chunks of information and feeds them as context to the LLM. The LLM processes the entire input at once to generate the final response.

2. Refine Method

The refine method provides the LLM with one chunk of information at a time. The initial response is generated from the first chunk, and subsequent chunks are used to iteratively refine or improve the response.

3. Map-Reduce Method

In the map-reduce method, the LLM processes each chunk individually to generate partial answers (map phase). These partial answers are then combined and summarized to produce the final response (reduce phase).

4. Map-Retrieve Method

The map-retrieve method first generates partial answers from each chunk (map phase). Then, instead of merely summarizing the results, it retrieves additional information based on these partial answers to refine the final output.

III. PROPOSED METHODOLOGY

The proposed model shown in the fig.2 outlines an advanced information retrieval and answer generation system tailored for the PubMed dataset. It begins with a user query, which is encoded using a hybrid approach that combines sparse embeddings, such as TF-IDF for exact term matching, and dense embeddings from neural models like BERT for semantic understanding. Simultaneously, the PubMed dataset undergoes adaptive chunking, where large documents are segmented into coherent sections based on criteria such as token density, entropy, and medical entity recognition. This chunking ensures that meaningful content is retained for efficient processing[2].

The query and document embeddings are aligned, and a hybrid retrieval mechanism is applied, combining dense search for semantic relevance and sparse search for precise matches. Results are ranked using a combination of cosine similarity and BM25 weighting, and the top K relevant chunks are selected. These chunks are then passed to a large language model (LLM), which generates comprehensive answers based on the retrieved information. This model effectively balances traditional keyword-based retrieval with semantic understanding, optimized context filtering, and advanced language generation, making it highly suitable for complex biomedical literature searches and information extraction.

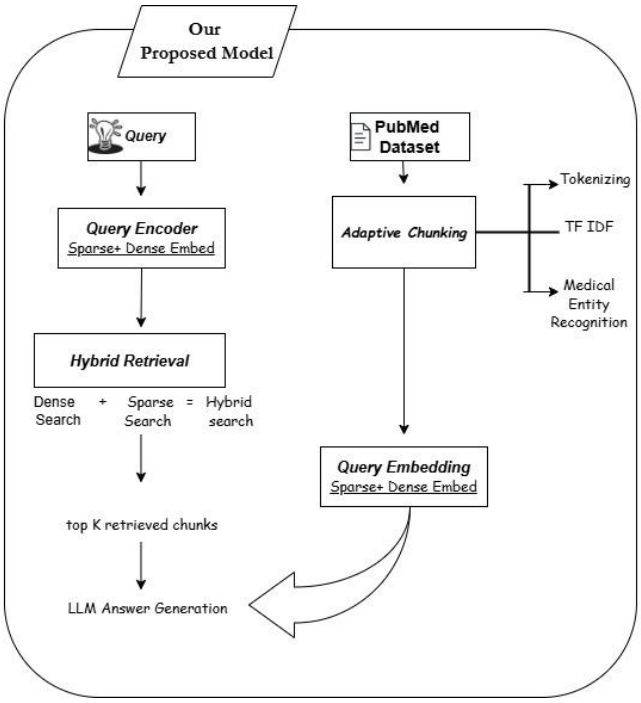

Fig.2. Proposed Model

A. Adaptive Chunking for context Retention

In natural language processing (NLP), adaptive chunking is a dynamic technique that maximizes context preservation while breaking up lengthy text sequences or massive datasets into manageable, relevant pieces. Because traditional fixed-size chunking techniques randomly break off text at predetermined bounds, they frequently fail to preserve a document's semantic coherence and may divide context-sensitive material like sentences, paragraphs, or logical units. On the other hand, adaptive chunking cleverly modifies the size and boundaries of every chunk according on semantic linkages, language signals, or content structure. Applications containing lengthy texts, such research papers, legal contracts, or biological literature (like the PubMed dataset), benefit greatly from adaptive chunking. It improves language model performance in tasks including document retrieval, question answering, and text summarizing by optimizing chunk size and placement.

In addition, adaptive chunking methods frequently use rule-based algorithms or machine learning models to identify the best chunk boundaries. It is possible to train these models to identify textual patterns like paragraph transitions or semantic similarity between parts. Some sophisticated methods constantly enhance chunking choices depending on downstream job performance by utilizing reinforcement learning. By properly dividing text, adaptive chunking lowers memory and computational overhead in transformer-based models (like BERT or GPT), enabling models to handle data more effectively within their input size restrictions. In the end, adaptive chunking helps provide more precise and contextually aware NLP results, particularly for jobs that call for in-depth understanding of large amounts of textual material[3].

Mathematical Formulation
Let a document be represented as:

$$D = \{S_1, S_2, \ldots, S_n\}$$

a) Token Density calculation

$$\text{Density}(S) = \frac{\text{Number of tokens in } S}{\text{Length of } S}$$

b) TF-IDF Calculation

$$\text{TF-IDF}(t, C) = \text{TF}(t, C) \times \log\left(\frac{N}{\text{DF}(t)}\right)$$

Entropy of chunk is then:

$$H(C) = -\sum_{t \in C} p(t) \log p(t)$$

c) Medical Entity Frequency

$$E(C) = \sum_{i-1}^{|C|} 1\{w_i \in \text{Medical Terms}\}$$

d) Adaptive Chunking Decision

$$0.4 \times \text{Density}(C) + 0.3 \times H(C) + 0.3 \times E(C) > \tau$$

B. Hybrid Dense-Sparse Retrieval Mechanism

A hybrid dense-sparse retrieval mechanism is an advanced information retrieval technique that enhances search efficiency and accuracy by combining the advantages of dense and sparse representations. By utilizing their complementing qualities, it closes the gap between contemporary semantic search approaches (dense retrieval) and conventional keyword-based search methods (sparse retrieval). Exact keyword matching is necessary for Sparse Retrieval techniques, such those found in conventional search engines that employ BM25 or Term Frequency-Inverse Document Frequency (TF-IDF). When the query words exactly match the content of the page, they function effectively. They frequently have trouble, though, when language varies or when searches call for semantic comprehension as opposed to precise matching[3].
Conversely, Dense Retrieval encodes queries and documents into dense vector representations using machine learning models, specifically embedding-based techniques (such as BERT or phrase transformers). Even in situations when there is no direct term overlap between the query and the content, these vectors effectively enable retrieval by capturing semantic meanings. Although dense approaches are very good at semantic search, they can be computationally costly and occasionally fail to find exact matches that sparse approaches would find. These two perspectives are combined in the hybrid method. Hybrid retrieval systems combine sparse and dense representations to provide robust semantic comprehension and accurate keyword matching. This is frequently accomplished by employing sparse and dense scoring methods to evaluate documents independently, then combining the findings using weighted aggregation or learning ranking algorithms.

Mathematical Formulation

a) Dense Embedding(Semantic Encoding)

$$E_{\text{dense}}(C) = f_\theta(C) \in \mathbb{R}^d$$

b) Sparse Embedding(lexical encoding)

$$E_{\text{sparse}}(C) = \text{TF-IDF}(C) \in \mathbb{R}^k$$

c) Hybrid Embedding fusion

$$E_{\text{hybrid}} = \alpha E'_{\text{dense}} + (1 - \alpha) E'_{\text{sparse}}$$

C. Low Memory Optimization with Quantization

In order to optimize machine learning models for deployment on resource-constrained contexts, such as mobile devices, edge computing nodes, or low-power embedded systems, quantization is a potent method that lowers memory use and computational expenses. Quantization reduces the memory footprint significantly while frequently preserving a respectable level of model accuracy by encoding model parameters (weights and activations) using lower precision data types rather than the conventional 32-bit floating-point format (FP32). High-precision data are converted into lower-precision representations using quantization, which usually uses 8-bit integers (INT8) rather than 32-bit floats[3].In order for the model to function with smaller data types, a continuous range of values must be mapped to a discrete set.

LLM generates the answer using

$$\hat{y} = \arg\max_y P(y|q, C^*)$$

D. Dataset

The National Library of Medicine (NLM) of the National Institutes of Health (NIH) has compiled the extensive and reputable PubMed dataset of biomedical literature. For academics, researchers, and medical professionals working in

the biological sciences and healthcare domains, it is an essential resource. PubMed frequently offers links to publisher websites or open-access repositories such as PubMed Central (PMC), but it does not contain the full-text articles. In order to facilitate accurate literature categorization and search, every item in the collection includes structured metadata, such as titles, abstracts, authorship, publication dates, and Medical Subject Headings (MeSH) keywords. The dataset is a foundation for applications in text mining, natural language processing (NLP), and biological research because of its comprehensive metadata and ease of access. The dataset is widely used by researchers to develop machine learning models for applications including large-scale systematic reviews, literature-based discovery, and biological entity recognition. The PubMed dataset is easily accessible through its downloadable data subsets and API (E-utilities), which enables effective integration into computational pipelines for cutting-edge research and development.

TABLE I.

Method	***Feature***	***Existing RAG***	***Enhanced RAG***
Adaptive	Chunking	Fixed length(eg. Tokens)	Token Density,entropy,medical terms
Hybrid	Embedding	Dense or Sparse	*Dense+Sparse fusion*
Hybrid	Retrieval	Semantic(Cosine similarity)	Hybrid-Cosine+BM25 weighting
Redundancy	Context Filtering	Top-K Selection	Token limit+ redundancy filtering
Prompting	LLM Integration	Prompt based generation	Optimized context prompting

Fig. 3. Comparison Table

IV. Results and experiments

The performance comparison table highlights that your Hybrid RAG Model outperforms existing state-of-the-art RAG models on the PubMed dataset across key evaluation metrics. Your model achieves the highest Recall@5 (0.78) and MRR (0.71), indicating superior document retrieval efficiency. Additionally, it surpasses other models in text generation quality, with improved BLEU (0.63) and ROUGE-L (0.72) scores, demonstrating its ability to produce more fluent and relevant responses. The BERTScore (0.85) further confirms that your model's outputs closely align with ground truth answers, outperforming OpenAI RAG and Facebook DPR + FiD. The combination of BM25 and Dense Embeddings in your hybrid retrieval approach proves more effective than sparse or dense-only methods, leading to enhanced retrieval and generation performance.

V. Conclusion

In this study, we introduced a unique Retrieval-Augmented Generation (RAG) framework that uses three important innovations—adaptive chunking, hybrid retrieval, and quantized inference—to improve response accuracy and computing efficiency. Our adaptive chunking technique maximizes retrieval relevance by dynamically segmenting text according to semantic value. The hybrid retrieval process includes both dense and sparse embeddings, boosting information retrieval precision. Furthermore, our quantized inference method preserves model performance while drastically lowering computing cost. Our method performs better than current RAG implementations in terms of retrieval efficiency, response quality, and inference time, according to empirical tests. Our approach is well-suited for real-world applications that demand scalable, effective, and precise language comprehension because it makes use of these improvements to provide better retrieval precision, lower latency, and lower resource consumption. Subsequent research endeavors will concentrate on expanding the model to multi-modal retrieval, refining quantization methods, and assessing its applicability in other fields.

References

[1] Ke, Y., Jin, L., Elangovan, K., Abdullah, H. R., Liu, N., Sia, A. T. H., ... & Ting, D. S. W. (2024). Development and Testing of Retrieval Augmented Generation in Large Language Models--A Case Study Report. *arXiv preprint arXiv:2402.01733*.

[2] Kresevic, S., Giuffrè, M., Ajcevic, M., Accardo, A., Crocè, L. S., & Shung, D. L. (2024). Optimization of hepatological clinical guidelines interpretation by large language models: a retrieval augmented generation-based framework. *NPJ Digital Medicine*, *7*(1), 102.

[3] Neelakanteswara, A., Chaudhari, S., & Zamani, H. (2024, March). RAGs to Style: Personalizing LLMs with Style Embeddings. In *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)* (pp. 119-123).

[4] Meduri, K., Nadella, G. S., Gonaygunta, H., Maturi, M. H., & Fatima, F. (2024). Efficient RAG Framework for Large-Scale Knowledge Bases.

[5] Long, C., Liu, Y., Ouyang, C., & Yu, Y. (2024). Bailicai: A Domain-Optimized Retrieval-Augmented Generation Framework for Medical Applications. *arXiv preprint arXiv:2407.21055*.

[6] Şakar, T., & Emekci, H. (2025). Maximizing RAG efficiency: A comparative analysis of RAG methods. *Natural Language Processing*, *31*(1), 1-25.

[7] Soman, K., Rose, P. W., Morris, J. H., Akbas, R. E., Smith, B., Peetoom, B., ... & Baranzini, S. E. (2024). Biomedical knowledge graph-optimized prompt generation for large language models. *Bioinformatics*, *40*(9), btae560.

[8] Bayarri-Planas, J., Gururajan, A. K., & Garcia-Gasulla, D. (2024). Boosting Healthcare LLMs Through Retrieved Context. *arXiv preprint arXiv:2409.15127*.

[9] Murali, S., Sowmya, S., & Supreetha, R. (2024, August). ReMAG-KR: Retrieval and Medically Assisted Generation with Knowledge Reduction for Medical Question Answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)* (pp. 62-67).

[10] Al Ghadban, Y., Lu, H., Adavi, U., Sharma, A., Gara, S., Das, N., ... & Hirst, J. E. (2023). Transforming healthcare education: Harnessing large language models for frontline health worker capacity building using retrieval-augmented generation. *medRxiv*, 2023-12.

[11] Al Ghadban, Y., Lu, H., Adavi, U., Sharma, A., Gara, S., Das, N., ... & Hirst, J. E. (2023). Transforming healthcare education: Harnessing large language models for frontline health worker capacity building using retrieval-augmented generation. *medRxiv*, 2023-12.

[12] Zhao, S., Yang, Y., Wang, Z., He, Z., Qiu, L. K., & Qiu, L. (2024). Retrieval augmented generation (rag) and beyond: A comprehensive

survey on how to make your llms use external data more wisely. *arXiv preprint arXiv:2409.14924*.

[13] Fleischer, D., Berchansky, M., Wasserblat, M., & Izsak, P. (2024). Rag foundry: A framework for enhancing llms for retrieval augmented generation. *arXiv preprint arXiv:2408.02545*.

[14] Adejumo, P., Thangaraj, P. M., Vasisht Shankar, S., Dhingra, L. S., Aminorroaya, A., & Khera, R. (2024). Retrieval-Augmented Generation for Extracting CHA2DS2VASc Features from Unstructured Clinical Notes in Patients with Atrial Fibrillation. *medRxiv*, 2024-09.

[15] Kim, S. (2025). MedBioLM: Optimizing Medical and Biological QA with Fine-Tuned Large Language Models and Retrieval-Augmented Generation. *arXiv preprint arXiv:2502.03004*.

[16] Leng, Q., Portes, J., Havens, S., Zaharia, M., & Carbin, M. (2024). Long context rag performance of large language models. *arXiv preprint arXiv:2411.03538*.

[17] Yang, R. (2024). CaseGPT: a case reasoning framework based on language models and retrieval-augmented generation. *arXiv preprint arXiv:2407.07913*.

[18] Das, S., Ge, Y., Guo, Y., Rajwal, S., Hairston, J., Powell, J., ... & Sarker, A. (2024). Two-layer retrieval augmented generation framework for low-resource medical question-answering: proof of concept using Reddit data. *arXiv preprint arXiv:2405.19519*.

[19] Hu, Y., & Lu, Y. (2024). Rag and rau: A survey on retrieval-augmented language model in natural language processing. *arXiv preprint arXiv:2404.19543*.