

LLM-Therapist: A RAG-Based Multimodal Behavioral Therapist as Healthcare Assistant

Fozle Rabbi Shafi
School of Computing
Queen's University
Kingston, ON
f.shafi@queensu.ca

M. Anwar Hossain
School of Computing
Queen's University
Kingston, ON
ahossain@queensu.ca

Abstract—Systems and services based on Large Language Models (LLM) are trending in the applied field of Artificial Intelligence (AI). The emergence of GPTs and subsequent improvements such as, unsupervised pre-training and transformer architecture have lead to its ability to generate human-quality text and understand context effectively. This has opened huge opportunity in healthcare domain to use AI-based assistant systems that can offer tailored recommendations and guidance for patients in numerous scenarios by considering multimodal data. This paper proposes LLM-Therapist as a multimodal personalized health care assistant for various types of patients. The proposed system uses Retrieval Augmented Generation (RAG) technique to improve the quality, accuracy, and relevance of generated response, which is specially important in providing healthcare assistance. We conducted experiments with LLM-therapist by extracting knowledge from domain-specific resources in mental health and patient's health data. Our experiments showed better efficiency and performance in providing personalized assistance.

Index Terms—Large Language Model, AI, Machine Learning, NLP, RAG, Health Care

I. INTRODUCTION

The evolution of AI in generating human-like text responses has seen significant advancements since the inception of the transformer model by Google [1], which marked a pivotal shift in language translation and text coherence capabilities. This groundbreaking work laid the foundation for subsequent developments in LLM, such as GPTs [2] [3], highlighting real-world applications of transformer-based neural networks. LLMs have since been widely adopted across various domains, demonstrating their versatility and powerful computational abilities. The integration of LLM-based technologies has the potential to transform the healthcare sector, revolutionizing user interactions and enhancing service delivery. These advancements lead to more personalized care, streamlined processes, and improved access to medical expertise, all of which can enhance patient outcomes and healthcare efficiency.

In healthcare, AI-driven chatbots and virtual agents have shown promising developments [4] [5], yet they often face challenges in providing personalized and contextually aware interactions. Traditional conversational agents have limitations in understanding complex contexts, maintaining natural conversations, and tailoring responses to individual needs, because they are usually trained on limited samples and can

capture simple patterns. However, the advent of LLM-based technologies [6] has begun to shift this dynamic, offering more sophisticated reasoning and comprehension akin to human interactions.

There are several available options for designing a LLM-based chat assistant for a healthcare scenario. Only a few well-established methods, such as few-shot prompting [7], fine-tuning [8] [9], and embedding [10], are widely recognized. Despite having existed for a few years, these methods are limited in many cases. While fine-tuning LLMs holds promise for any solution, its limitations can be substantial. A major concern is the typical nature of the fine-tuned model. The process acts as a "black box", making it difficult to comprehend the reasoning behind the LLM's outputs. This lack of transparency hinders our ability to assess the model's reliability and trustworthiness.

RAG [11] [12] offers a compelling alternative to traditional fine-tuning for many systems working with LLMs. Unlike fine-tuning [13], which locks the LLM into a specific knowledge-base through retraining, RAG leverages retrieval techniques to access and integrate up-to-date information directly during generation. This ensures the model's responses remain current with the latest knowledge, a significant advantage over the static data dependency of fine-tuning. Additionally, RAG excels at handling multimodal data sources, be they documents, structured databases, or unstructured information. This flexibility makes it suitable for a wider range of tasks compared to the domain-specific nature of fine-tuning.

Our research introduces a novel approach by proposing a multimodal LLM-based chat assistant specifically designed for the healthcare sector. This assistant not only leverages patient data sources but also incorporates advanced RAG techniques to enhance the personalization of responses. In RAG, user prompts are altered after retrieving relevant documents to create a personalized prompt. Our solution also incorporates function calling [14] with RAG technique to enable real-time data retrieval for interactive responses. By integrating diverse data sources and sophisticated retrieval mechanisms, our model aims to overcome the existing limitations of AI chatbots, providing more accurate, context-aware, and personalized patient support.

The remainder of this paper is organized as follows. Section

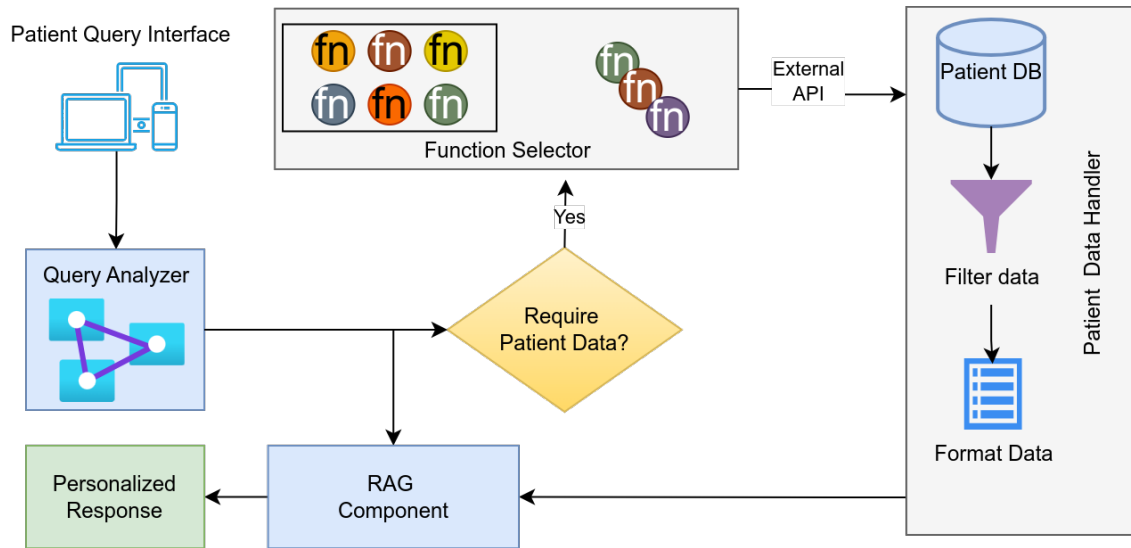


Fig. 1: System architecture showing how patient information will be collected when the patient is having a conversation with our system.

II comments on some relevant literature, Section III provides the details of proposed methodology, followed by Section IV that describes the implementation. Section V elaborates the experiment and result of this study. Finally, the paper concludes in Section VI with some future directions.

II. LITERATURE REVIEW

In this section, we outline relevant literature focusing on AI-driven healthcare chatbots to understand the gaps and requirements for designing and developing an improved personalized chatbot.

The study in [4] evaluated the acceptability of three digital sexual and reproductive health (SRH) services among 257 patients in Hampshire, UK: video consultations, live webchats, and AI-enabled chatbots. Findings reveal a preference for face-to-face initial contact, with video consultations and webchats acceptable for ongoing care. However, AI chatbots were less favored. Key influencers included age, ethnicity, previous STI history, and smartphone use. The study emphasizes the importance of maintaining human interaction in digital SRH transformations.

In the area of chatbot, the AI-powered medical chatbot [5] aims to improve healthcare access and reduce costs by offering preliminary disease diagnosis and information. It uses natural language processing and techniques like n-gram, TFIDF, and cosine similarity for accurate responses. Complex queries are handled by a third-party expert system, enhancing initial medical consultations efficiently. Similar solutions are reported in [15].

The "CataractBot" study [12] shows how this chatbot, developed with an eye hospital in India, uses AI and expert verification to provide accurate, multilingual information about cataract surgery, improving information reliability in an era of widespread misinformation. A trial with 49 participants demonstrated its effectiveness in making health information

accessible and trustworthy, thereby easing the burden on healthcare professionals and meeting diverse patient needs. Comparatively, both CataractBot and a general AI medical chatbot [5] enhance access to healthcare information through AI. While the general chatbot broadly diagnoses diseases, CataractBot offers specific, expert-verified cataract surgery information, also focusing on building trust with expert validation.

The study [16] improves interactions with Retrieval-Augmented Generation (RAG)-based agents by developing a suggestion question generator that uses dynamic contexts, such as few-shot examples and retrieved information. Experiments show this approach generates better questions than other methods, helping users more effectively communicate with the system. RAG based solutions are explored for developing medical assistant for infectious disease [17].

The REALM framework [18] enhances clinical predictions by integrating multimodal Electronic Health Records (EHR) data with external knowledge graphs. It uses a Large Language Model (LLM) to process clinical notes and a GRU model for time-series data, focusing on accurate entity extraction and alignment with medical standards. This integration is achieved through an adaptive multimodal fusion network, ensuring consistency and reducing errors. Tested on MIMIC-III mortality and readmission tasks, REALM significantly outperforms traditional models, demonstrating its effectiveness in refining clinical insights.

Each approach tackles different aspects of healthcare digitization with AI. While studies like [4] and [12] focus on the user acceptability and specific informational needs, technologies in [5] and [16] aim to enhance the efficiency of medical consultations and user interactions with AI. The REALM framework [18], on the other hand, illustrates the advanced application of AI in processing and integrating com-

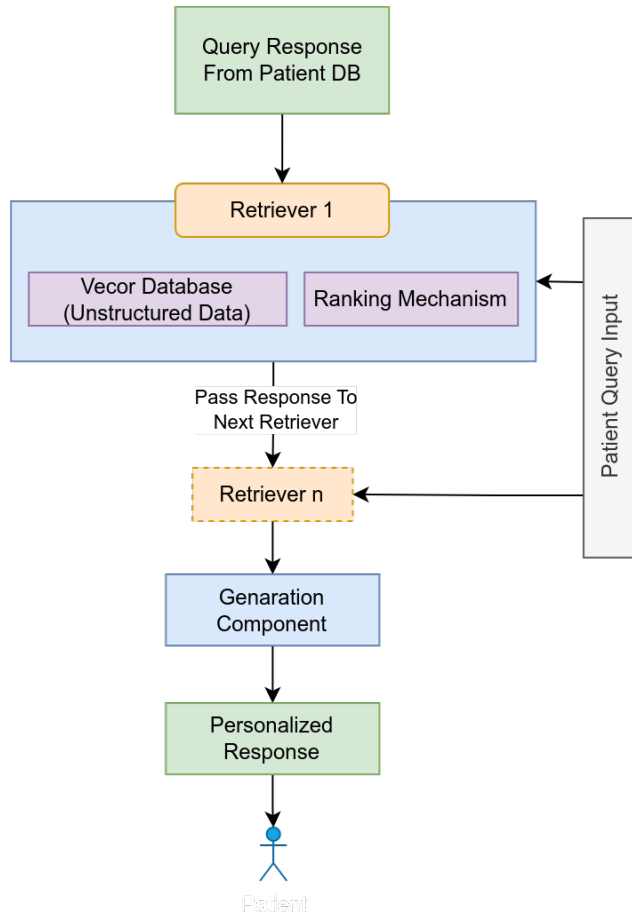


Fig. 2: The patient query is going through a series of processes, starting with function calling and then RAG. There might be more than one RAG based on necessity.

plex healthcare data for better clinical outcomes. Collectively, these studies underscore the diverse potentials and challenges of AI in healthcare, from enhancing patient interaction to optimizing clinical decision-making processes.

III. PROPOSED LLM-THERAPIST

This section elaborates the detail methodology of our proposed approach. We first highlight the system architecture of the LLM-Therapist system followed by the Knowledge-base development mechanism used in the system. Finally, we provide a system workflow to expose how the integrated system works.

A. System Architecture

The high-level building blocks of our proposed LLM-Therapist system is given in Fig. 1. The primary functional components are Query Analyzer, Function Selector, Patient Data Handler, and RAG component. We describe these components in the following.

1) *Query Analyzer*: This component receives user prompts written in natural language and extracts the relevant parameters to determine whether patient-related personalized data

need to be incorporated within the query response. The system uses function calling to facilitate this. A copy of the user query is passed to the RAG component irrespective of function calling to prepare the user response.

2) *Function Selector*: Based on the extracted parameters by the Query Analyzer, the Function Selector decides which functions match the user query parameters. This is facilitated by the function calling features of the respective LLM platforms, such as OpenAI. Accordingly, one or more External API call will be made. This allows the system to answer questions by querying real-time data sources, going beyond the limitations of their internal knowledge base. The LLM-Therapist accordingly can tailor its responses and recommendations to individual needs. This personalization is crucial in healthcare, where a one-size-fits-all approach often falls short.

3) *Patient Data Handler*: This component responds to API calls for preparing patient-specific data that relates to the corresponding query. The patient-specific data comes from Patient DB, which is a structured database prepared on patient's electronic health record, real-time sensory data, labs reports, and other relevant information.

4) *RAG component*: Fig. 2 shows the internal processes of the RAG component. In this stage, the user query and corresponding patient data is made available to this component. The Retriever 1 in Fig. 2 uses this data along with the knowledge base available in vector database for handling unstructured data. The creation of knowledge base is explained in Section III-B. In order to obtain more refined output, multiple retrievers are used in sequence. In our system, we adopted a hybrid approach, utilizing both function calling and RAG for a more precise response. Finally, the generation process prepares a fully personalized response to satisfy the user query.

B. Knowledge-base development

In order to effectively function and provide personalized response to a user, data preparation plays a vital role. The data comes from multiple sources, such as books, articles, news, media, etc. Our goal is to build a vector database based on these multi-source data. This process follows several steps. We start by extracting text from the given document, split the text into smaller chunks, and convert text chunks into vector embeddings using a pre-trained model, such as Sentence-Transformer. We then store the embeddings in a vector database, such as ChromaDB. While saving the data in vector database, we save relevant keywords as metadata to provide some context to the chunks. This is highly significant as it enhances the process of indexing for semantic search.

We also prepared a set of question-answer pairs based on selected number of chunks to pre-train our model using fine tuning mechanism. The pre-train process can be repeated multiple times to enhance its capability in specific domain. Fig. 3 illustrates the fine-tune approach we adopted for data preparation.

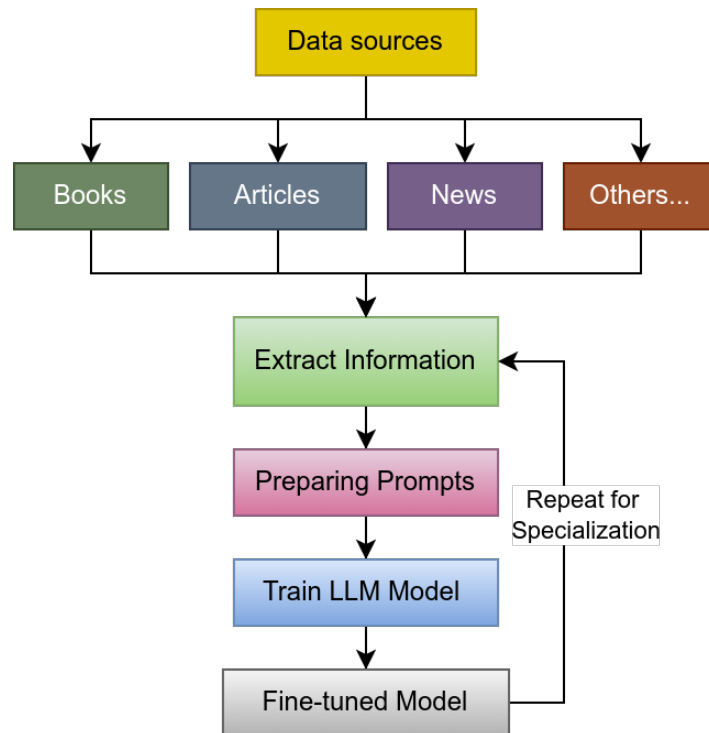


Fig. 3: Fine-tuning LLM by creating custom, high quality datasets.

C. System Workflow

At first, after receiving a patient's query, our system starts analyzing whether patient data needs to be fetched from an external API or not. In cases where a registered user uses our system, we store their medical history in our database. In the next phase, our system formats the response it gets from the external API and passes it to the next retriever, where it looks for relevant information about the query. As there is metadata associated with saved vectorized documents, our system can find the most relevant information related to the query. Finally, our system will format the patient's initial query with both patient data and information fetched from the vector database are passed to LLM to generate a highly personalized response.

IV. PROOF OF CONCEPT

To convert our system architecture into a software program, our system adheres to a series of sequential software processes. We implemented the processes in Python platform. A link to the implementation packages is available through: <https://github.com/devshafi/llm-therapist>

In our implementation of LLM-Therapist, We used gpt-3.5-turbo-0125 for fine-tuning personalized LLM model. Open AI's function calling is used in our solution. For Embedded Functions, we used SentenceTransformer and "all-MiniLM-L6-v2" as embedded model. For vector DB, ChromaDB is used in this version. The tools and technologies that are used in our implementation can be changed to newer or alternative packages.

V. EXPERIMENTS AND RESULTS

In order to demonstrate the functionality and performance of the developed LLM-Therapist system, we conducted an experiment with a hybrid methodology that involved utilizing both function calling and RAG to create a customized mental health chat assistant.

We started experimenting with fine-tuning-based approach. To generate the training dataset, we collected various mental health related books, articles, and news stories and prepared almost 500 pairs of instructions in the format of "system", "user", and "assistant". What we found is that this approach is good for an LLM-based system that is already trained on certain things. Major constraints in this regard include the preparation and curation of high-quality training datasets, defining the objective for fine-tuning, and providing the necessary computational resources.

To simulate the function calling approach, we gathered several thousand patient data points and stored them in a NoSQL database. At first, our attention was directed towards the essential indicators of patients, i.e. blood pressure, body temperature, heart rate, and oxygen saturation level, which were derived from a standardized dataset obtained through the use of the VitalDB at Seoul National University Hospital in Seoul, Korea [19]. Subsequently, we devised six discrete functions that can be invoked using the OpenAI "function calling" capability. Once the GPT model finished processing and provided a function call instruction with all the necessary arguments, we utilized our external API to retrieve data from the database regarding their vital signs. This simula-

```
query = "I am feeling very stressed and
anxious. How can I calm myself down?"
```

Fig. 4: Sample user query to LLM-Therapist and other platform.

```
Patient Data: {'name': 'Fozle Rabbi Shafi', 'age':
30, 'heart_rate': 73, 'blood_pressure': '120/80',
'body_temperature': '98.6 C', 'weight': '140 lbs',
'oxygen_saturation': '98%'}

Answer: Maybe even say out loud, "Stop," "Relax,"
or "REST ," just to remind yourself to not react
so quickly and automatically Then take a few slow
breaths to help yourself calm down before choosing
an alternative meaningful action Evaluate Next,
ask yourself what's happening in this situation
```

Fig. 5: Sample relevant retrieved data from patient DB and vector DB.

```
"It seems like you're feeling stressed and anxious,
Fozle Rabbi Shafi. One way to calm yourself down is to
take a few slow breaths and remind yourself to relax
before reacting. You can also try to evaluate the
situation and consider alternative actions to manage
your stress."
```

Fig. 6: Sample personalized response from LLM-Therapist.

tion demonstrates the effectiveness of the "function calling" mechanism in dynamically selecting the appropriate function to execute.

Lastly, in the multi-source approach, at first we captured the medical history data from function calling and then fed the response to the next phase while generating a response using LLM. With the combination of previously received vital signs from a patient and relevant suggestions and recommendations extracted from the vector database using the user query, the LLM provides highly personalized assistant support for a patient query.

For our experiments, we prepared different question sets, each consisting of questions based on the patient's requirements. We throw the questions at each of the existing chatbots and calculate the cosine similarity between the questions and the final response given by each platform. As our system can get patient histories using function calling and related information using RAG, it shows significant performance by a good margin compared to existing solutions.

Figure 4 presents a sample user query, followed by the relevant data retrieved from the patient database and vector database, as shown in Figure 5. Finally, Figure 6 illustrates a sample response generated by the LLM-Therapist.

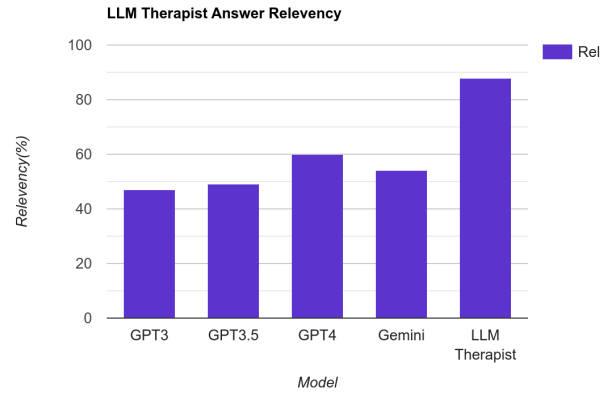


Fig. 7: Relevancy score of LLM-Therapist.

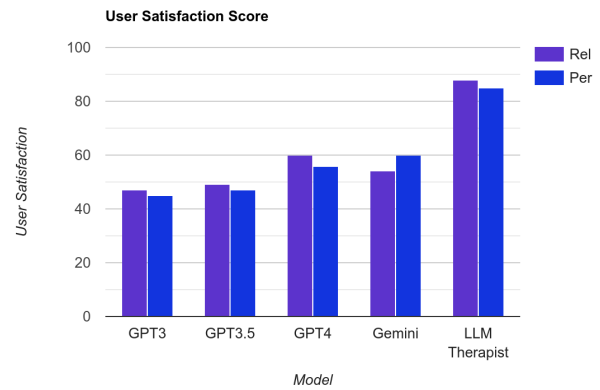


Fig. 8: Comparison of user satisfaction.

Based on our experiment, we also obtained the relevancy score of our health assistant as shown in Fig. 7. This reflects significant performance compared to several existing solutions.

We conducted a mini user studies with the developed LLM-Therapist in terms of user satisfaction. Fig. 8 shows result, which demonstrates the viability of our system.

VI. CONCLUSIONS AND FUTURE WORK

This research introduces LLM-Therapist, a chat assistant that combines Retrieval-Augmented Generation (RAG) with real-time function calls to enhance healthcare assistance. This approach personalizes interactions through a user-specific RAG system, tailoring experiences to each patient's medical history and preferences. Additionally, real-time function calls ensure access to the latest information, offering patients the most relevant guidance. By empowering patients and alleviating the workload on healthcare providers, this system demonstrates its potential to improve healthcare services. Future research will focus on deeper integration with healthcare ecosystems and evaluating its impact on patient outcomes through large-scale user studies. Ensuring the privacy and

security of patient-related data will also be a key priority, exploring solutions designed to safeguard sensitive information.

REFERENCES

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need.(nips), 2017," *arXiv preprint arXiv:1706.03762*, vol. 10, p. S0140525X16001837, 2017.
- [2] K. Ethayarajh, "How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings," *arXiv preprint arXiv:1909.00512*, 2019.
- [3] L. Floridi and M. Chiriatti, "Gpt-3: Its nature, scope, limits, and consequences," *Minds and Machines*, vol. 30, pp. 681–694, 2020.
- [4] T. Nadarzynski, O. Miles, A. Cowie, and D. Ridge, "Acceptability of artificial intelligence (ai)-led chatbot services in healthcare: A mixed-methods study," *Digital health*, vol. 5, p. 2055207619871808, 2019.
- [5] L. Athota, V. K. Shukla, N. Pandey, and A. Rana, "Chatbot for healthcare system using artificial intelligence," in *2020 8th International conference on reliability, infocom technologies and optimization (trends and future directions)(ICRITO)*, pp. 619–622, IEEE, 2020.
- [6] H. Raval, "Limitations of existing chatbot with analytical survey to enhance the functionality using emerging technology," *International Journal of Research and Analytical Reviews (IJRAR)*, vol. 7, no. 2, 2020.
- [7] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM computing surveys (csur)*, vol. 53, no. 3, pp. 1–34, 2020.
- [8] B. Zhang, Z. Liu, C. Cherry, and O. Firat, "When scaling meets llm finetuning: The effect of data, model and finetuning method," *arXiv preprint arXiv:2402.17193*, 2024.
- [9] K. W. Church, Z. Chen, and Y. Ma, "Emerging trends: A gentle introduction to fine-tuning," *Natural Language Engineering*, vol. 27, no. 6, pp. 763–778, 2021.
- [10] R. Patil, S. Boit, V. Gudivada, and J. Nandigam, "A survey of text representation and embedding techniques in nlp," *IEEE Access*, vol. 11, pp. 36120–36146, 2023.
- [11] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, and H. Wang, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, 2023.
- [12] P. Ramjee, B. Sachdeva, S. Golechha, S. Kulkarni, G. Fulari, K. Murali, and M. Jain, "Cataractbot: An llm-powered expert-in-the-loop chatbot for cataract patients," *arXiv preprint arXiv:2402.04620*, 2024.
- [13] Wikipedia, "Fine-tuning (deep learning)." Last access: April 10, 2024.
- [14] C. Jarvis and J. Palermo, "How to call functions with chat models," 2023. Last access: April 15, 2024.
- [15] C. Wu, Z. Lin, W. Fang, and Y. Huang, "A medical diagnostic assistant based on llm," in *China Health Information Processing Conference*, pp. 135–147, Springer, 2023.
- [16] A. Tayal and A. Tyagi, "Dynamic contexts for generating suggestion questions in rag based conversational systems," in *Companion Proceedings of the ACM on Web Conference 2024*, pp. 1338–1341, 2024.
- [17] M. Raja, E. Yuvarajan, et al., "A rag-based medical assistant especially for infectious diseases," in *2024 International Conference on Inventive Computation Technologies (ICICT)*, pp. 1128–1133, IEEE, 2024.
- [18] Y. Zhu, C. Ren, S. Xie, S. Liu, H. Ji, Z. Wang, T. Sun, L. He, Z. Li, X. Zhu, et al., "Realm: Rag-driven enhancement of multimodal electronic health records analysis via large language models," *arXiv preprint arXiv:2402.07016*, 2024.
- [19] H.-C. Lee, Y. Park, S. B. Yoon, S. M. Yang, D. Park, and C.-W. Jung, "Vitaldb, a high-fidelity multi-parameter vital signs database in surgical patients," *Scientific Data*, vol. 9, no. 1, p. 279, 2022.