The 10th International Conference on Digital Arts, Media and Technology (DAMT)
and 8th ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (NCON)

2025 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON) | 979-8-3315-4327-3/25/$31.00 ©2025 IEEE | DOI: 10.1109/ECTIDAMTNCON64748.2025.10961967

Developing a RAG Agent for Personalized Fitness and Dietary Guidance

Min Swan Pyae, Sai Sithu Phyo,
Saw Thomas Maung Maung Kyaw, Thet Swe Lin, Nacha Chondamrongkul
Computer and Communication Engineering for Capacity Building Research Center,
School of Applied Digital Technology, Mae Fah Lunag University.
Chiang Rai, Thailand.
6531503155@lamduan.mfu.ac.th, 6531503179@lamduan.mfu.ac.th
6531503181@lamduan.mfu.ac.th, 6531503189@lamduan.mfu.ac.th, nacha.cho@mfu.ac.th

***Abstract*— This paper focuses on implementing and evaluating a Retrieval-Augmented Generation (RAG) system that provides personalized fitness and dietary guidance by combining real-time data from wearable devices with curated knowledge sources. While the current implementation depends on a carefully selected set of PDF files as a source of knowledge base, extracting relevant information in order to frame the answer to a question by the user, the planned integration of a curated, well-structured database aims to enhance the system's reliability and scope. The paper presents an overview of RAG-based system-related literature, describes the process of implementing the RAG component, and discusses the effectiveness in generating relevant and accurate answers. This implementation demonstrates the capability of RAG technology to develop interactive and effective solutions for dietary and fitness applications and sets a foundation for further system improvements.**

Index Terms*—Retrieval-Augmented Generation, RAG, fitness, dietary

I. Introduction

In the past few years, the evolution of Artificial Intelligence (AI) has reached a point where it leads to advancements in various domains including the nutrition and fitness sector. One of the most impactful improvements of AI is the integration of Large Language Models (LLMs) with the external data through Retrieval Augmented Generation (RAG). Even though the traditional LLMs can generate coherent texts, they hallucinate, contain factual inaccuracies and lack the ability to generate in-depth and reliable answers under a specific field. RAG can tackle this issue by using external retrieval mechanism that takes in reliable and up-to-date data sources. This not only reduces hallucinations, but also improves reliability and accuracy of responses, especially in the fields that require precise and in-depth knowledge.

This paper studies the implementation of RAG in the health industry that can generate and guide the users to have tailored advice on their diet and nutrition. By analyzing the input data from wearable devices and referencing the reliable data, this RAG system generates nutrition needed for different individuals. The core architecture of the system includes a vector store for document retrieval and a generative model for response execution. Additionally, it also includes text-to-speech and voice recognition features that lead to seamless user interactions. However, key issues such as reliance on limited datasets, hallucinations, and voice recognition performance require further development. This paper discusses the steps taken to implement a RAG-based system for personalized fitness and dietary guidance and outlines plans to address these limitations through curated databases, noise handling, and system optimization.

Our project focuses on developing a system that adapts to the biometric data of the users, their different goal settings and generate personalized nutritional advice based on them. Individualized recommendations are crucial because every person's dietary needs, health conditions, and fitness objectives vary significantly. A generic approach often fails to address these unique needs of each person, which can result in less effective advice or even cause health problems. By providing tailored guidance, the system ensures that recommendations align with each user's specific requirements, promoting better adherence to healthier habits and more effective outcomes. The relevance and reliability of the system has increased when RAG technology is integrated to it which leads to a more effective health management tool in this digital age.

II. Related Works

A. Retrieval Augmented Generation (RAG)

RAG (Retrieval Augmented Generation) is an approach to alleviate some of the weaknesses and suffering large language models have such as domain knowledge gaps, factuality issues, and hallucinations. The domain knowledge gaps occur when models are not trained on specific topics, usually affecting their accuracy. This carries over to factuality issues because the model could be giving wrong or outdated information derived from obsolete training data. Hallucinations are when the model generates false but plausible statements. LLMs are based on language patterns, not actual understanding, so this is inescapable. It becomes a particularly thorny problem since it's hard to recognize and tends to engender misplaced trust. Retrieval Augmentation Generation, or RAG for short, provides relevant and up-to-date information and improves accuracy to decrease hallucinations. RAG helps knowledge-intensive tasks that need frequent updates. Proposed by Meta AI authors in [1], it's a model that combines language generation with external retrieval systems, allowing models to have access to the latest data for generating text. The three core components of RAG

are Retrieval, Augmentation, and Generation. Retrieval fetches task-related information from a large database. Texts get chunked for different models to be fine-tuned for better domain-specific understanding. Queries and embeddings get reformulated to align with document semantics. The final step is the alignment of retriever outputs to LLM preferences. Augmentation adds context to the user query, enriching it with relevant chunks. That ensures LLMs have necessary background information. In the Generation step, the model uses the augmented input to generate a response. It crafts a coherent reply informed by the context. Feeding recovered chunks into generation reduces hallucinations and errors. [2] RAG has evolved from Naive to Advanced and Modular. Naive RAG retrieves and generates sequentially, while Advanced RAG ensures factual accuracy and context. The most advanced ones include fine-tuning the retrieval model on certain data and query rewriting. Modular RAG separates retrieval from generation for better optimization. It may use different retrieval methods with various language models, so it can be widely used. Retrieval Augmentation of Generation is a technique that uses external knowledge for improving accuracy and lowering hallucinations of Large Language Models to increase the reliability of specialized fields and extend the applications of LLMs. RAG revolutionizes our way of interacting with AI and information retrieval.

B. Lifestyle Recommendation

Artificial intelligence is transforming healthcare by applying advanced analytics techniques like machine learning and natural language processing to both structured and unstructured data. Nowadays, healthcare is improving due to the rapid evolution of conversational AI, especially Large Language Models (LLMs) in areas like diagnostic imaging, diabetic retinopathy and lifestyle improvements. The performance of Large Language models (LLMs) in healthcare chatbot applications is greatly improved by combining with Retrieval-Augmented Generation (RAG), highlighting the superiority of conventional question-answering techniques over generative approaches [3]. SouLLMate, an adaptive LLM-driven system that leverages advanced technologies like RAG and prompt engineering to enhance mental health support through features such as risk detection and proactive guidance dialogue. The SouLLMate system has many benefits including increasing patient intake capacity in underserved areas, automating tasks and improving efficiency but performance issues during extended interactions and restrictions in model selection that may limit research scope [4]. Adding specific healthcare data to LLMs increases response reliability, particularly when answering questions about chronic diseases. The augmented Llama model performs well for shorter responses, while ChatGPT 3.5 excels with longer output, and integrating both models improves assessment scores, demonstrating the efficiency of RAG systems in healthcare. There are also limitations including issues with language specificity, dataset size, scalability, reliance on external knowledge sources, ethical concerns and risk of hallucinations, highlighting the need for further exploration to enhance their applicability and reliability of LLMs [5]. In recent studies [6], virtual assistants tailored for older adults have demonstrated significant potential in improving lifestyle by offering individualized, contextually relevant support. Despite its emphasis on ethical data handling and scalable features, the initiative faces issues such as technological limitations, potential inaccuracies and the requirement for accessibility with varying levels of tech sophistication, which may hinder widespread adoption and effectiveness. According to [7], the characteristics and effectiveness of AI chatbot interventions on physical activity, healthy eating, and weight management finding that chatbots have the potential to increase physical activity although there is still little and conflicting data regarding their effects on diet and weight management. Additionally, the research in [8] shows that iDISK2.0-RAG system significantly improves the accuracy and reliability of dietary supplements information retrieval by integrating a comprehensive knowledge base with a user-friendly interface, achieving over 95% accuracy in responding True/False and multiple-choice questions. However, because of the limitations in mapping software and the quality of the source data, some residual errors still exist, and the system's efficacy depends on constant updates and enhancements to guarantee accuracy. The work in [9] proposes a framework that integrates Knowledge Graphs (KGs) and Causal Graphs (CGs) with LLMs to enhance personalized healthcare applications by addressing challenges related to accuracy, trustworthiness and personalization. However, it faces limitations such as the inability to bridge structured knowledge with LLMs, high computational costs, challenges in querying large graphs, the need to keep knowledge current, and the need to modify LLMs for domain-specific applications without sacrificing their general adaptability.

III. Lifestyle Recommendation System

The Related Works section highlighted the strengths and limitations of current AI-driven lifestyle recommendation systems; solutions like SouLLMate and iDISK2.0-RAG show the potential of RAG for improving mental health support and accuracy in dietary recommendations. However, research shows issues such as performance drops during extended interactions, static data reliance, and accessibility challenges are still pervasive. Our RAG-based lifestyle recommendation system compensates for these weaknesses while building on their strengths. It takes advantage of up-to-date information from wearable devices combined with the most advanced retrieval mechanisms to make sure that the recommendations made are dynamic and tailor-made. Moreover, the database, when updated continuously, holds fewer errors and enhances accuracy; hence, it's greatly more adaptable, scalable, and reliable than ever before.

A. Recommendation Mechanism

The RAG system for lifestyle recommendation includes the integration of wearable devices, real-time data streaming service, database, user interface and the RAG itself. Wearable gadgets track and send the user data to the database through data streaming service like Apache Kafka, and those data are retrieved later to provide to RAG so that it can generate personalized recommendations depending on individual user. Finally, those generated plans will be delivered to the user through an interface. The detailed description of RAG agent is illustrated in the figure below, showing how the components inside it interact with one another and cooperate to generate recommendations for users.

The recommendation mechanism of this Retrieval Augmented Generation (RAG) is quite simple. Firstly, the RAG agent will be provided with a database that has embedded data about nutrition and workout inside. After it receives the query (question) from the user, it will find and retrieve the data related to the input and generate answer according to what it found in the provided data. If it cannot find anything related in the database, it will return messages such as it does not have access to such kind of knowledge.

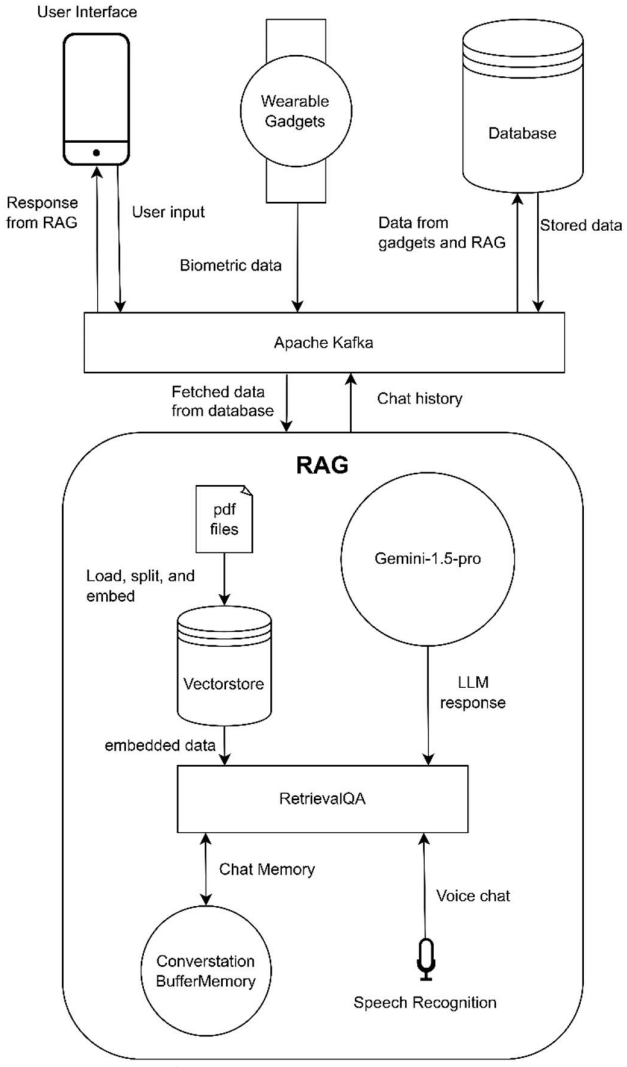

Fig 1. Architecture Overview: Component Interactions

B. Core Algorithm

The recommendation mechanism for the RAG agent is designed for interactive query handling using both text and audio inputs. The system first goes through setup, including all configurations: setting up the audio playback, adding environment variables, and an API key to be used for integration with the Google Generative AI model. It maintains a vector store where it preloads existing document embeddings; otherwise, it generates embeddings from PDFs. A retriever fetches the documents most relevant to a given query, while a large language model further processes the query to generate evidence-based answers. A custom memory handler logs and retrieves conversation history to maintain context. The agent aggregates the retrieved document content, user queries, and conversation history into a structured prompt for the LLM. It also incorporates text-to-speech and speech recognition functionalities to enable the user's interaction in both audio and text modes. The main function ties these components together, allowing users to choose their preferred input method, interact with the agent, and receive detailed responses augmented with source citations. The following pseudocode contains the main algorithm of the RAG agent:

1. Initialize audio mixer and load API key from environment.
2. Set up vector store: if existing, load it; if not, load PDFs, split into chunks, and create embeddings.
3. Initialize retriever for document retrieval and configure LLM with specified model and parameters.
4. Create a custom memory handler to log and retrieve past conversations.
5. Set up RetrievalQA with retriever and LLM for answering user queries.
6. Define `call_rag_agent` to:
   - Retrieve relevant documents.
   - Construct a prompt using retrieved text and conversation history.
   - Get response from LLM, format it with sources, and save to memory.
7. Implement text-to-speech and speech recognition functions for handling input and output.
8. Define `main` function to allow user choice between audio and text input modes:
   - For audio, listen and respond in a loop.
   - For text, read, process, and respond in a loop.
9. Run `main` to start the program and handle user queries interactively.

C. Prompts

In a RAG system, prompts act as instructions that tell the model what information to focus on when generating recommendations. For this RAG system, we use a customized prompt to make the system generate responses about nutrition based on the context (currently PDF files, which will later be replaced with well-curated database) provided in the database. The customized prompt we applied to the RAG is as follows:

"You are an AI expert specializing in nutrition and fitness, providing personalized dietary and fitness recommendations. Answer user queries concisely, relying strictly on the provided knowledge base. Cite sources where relevant and avoid speculative statements."

This instruction-based prompting guides the system to retrieve relevant information and provide a response tailored to the user's specific questions about nutrition and meal plans. In addition to the customized system prompt, the RAG system was tested using various user query prompts to evaluate its performance and accuracy. Examples of user queries include:

- *" What kind of foods are good for weight loss? "*
- *" What vitamins and minerals are essential for improving immunity, and how can they be incorporated into a diet? "*
- *" I am a 25-year-old female, weighing 140 pounds, with a sedentary lifestyle. What dietary changes should I make to lose weight safely? "*
- *" What are some general nutritional guidelines for people recovering from an illness? "*
- *" What are the best pre-workout and post-workout meals for endurance athletes? "*

These evaluation prompts were designed to test the system's ability to retrieve relevant nutritional data from the database and provide concise, evidence-based recommendations. The combination of the customized system prompt and diverse user queries ensured the RAG model delivered precise and context-specific answers to meet user needs.

D. Implementation

The Retrieval Augmented Generation (RAG) agent we have developed is a speech-driven AI assistant, designed to communicate with users through text and voice recognition, integrating advanced natural language processing techniques like text-to-speech library, speech recognition module and a Large Language Model (LLM). This solution integrates cutting-edge components such as Google GenerativeAI for LLM, LangChain for orchestration, Chroma for vector storage and document loaders for data ingestion to provide precise context-aware responses in fitness and nutrition data scenarios by processing voice input and delivering synthesized speech responses. Google Generative AI package is the core language model used with the Gemini-1.5-Pro model which provides sophisticated, contextually aware responses, fine-tuned with a temperature setting of 0.7 for a balanced blend of creativity and coherence. LangChain Framework acts as the main orchestrating tool linking the generative model with document loaders, memory management and retrieval mechanisms. The Chroma Vector Store serves as a database to store and retrieve embeddings generated from input documents. Document Loaders parses and loads data from PDF documents stored in directory using PyPDFLoader and DirectoryLoader. Google Generative AI Embeddings generate high-quality embeddings that assist in precise document retrieval. File-Based memory, a custom memory saves and loads conversation history in a JSON format, retaining a memory of past interactions. The Speech Recognition Module uses Google's speech-to-text API to convert spoken words into text using the speech_recognition library. The Text-to-Speech (TTS) Module uses the gTTS library to convert text responses into speech, using the pygame.mixer module for auditory feedback. For the knowledge base, 10 PDF files about dietary and exercise are stored in a folder.

The system initializes by loading environmental variables using dotenv to extract the API key for Google Generative AI. Next, it receives data which is parsed PDF documents from the knowledge base folder using DirectoryLoader and PyPDFLoader and then uses RecursiveCharacterTextSplitter to split the documents into 1000-character chunks for optimized retrieval. Currently, only a few pdf files are stored to test the system and later, we are going to upgrade the database with organized and formatted fitness and nutrition plans. The document embeddings are generated using GoogleGenerativeAI-Embeddings and stored in Chroma, which loads an existing vector database if available to avoid redundant processing. When a user speaks to the microphone, the system records the speech, transforms it into a digital audio signal and sends it to the Speech Recognition module, which uses Google's recognition API to turn the audio into text. The RAG agent receives the text, searches the Chroma vector store for relevant documents. Then, Chroma vector store retrieves the top 5 most relevant documents chunks which are passed to the LLM through the RetrievalQA chain. The LLM synthesizes these chunks with the conversational context to produce a context-aware response, ensuring the relevance of the nutrition data provided. Then the response is generated by using the Google Gemini model by integrating the retrieved information and conversation history. The generated text is formatted for clarity and converted into speech using the gTTS library with the output played back to the user through pygame's mixer module.

Improvements on the RAG system include development of the knowledge base, memory handling, voice recognition, retrieval fidelity, and the conversation context. The knowledge base shall also have incorporated APIs such as FoodData Central; it shall classify data for the purpose of personalization. Knowledge updates will be facilitated using LangChain's DataLoader APIs. Dynamic retention of contexts, with auto-clear features, will be allowed by migrating to LangChain's Conversation Buffer Memory. Voice recognition will be powered with noise suppression tools, for instance, PyAudioEffects, VAD (Voice Activity Detectors) utilities, and custom acoustic models, while having offline fallbacks such as VOSK (speech recognition toolkit). Retrieval fidelity will be enhanced through hybrid approaches combining semantic and keyword-based methods, dynamic k optimization, fine-tuned embeddings, and cross-encoder re-rankers for relevance and accuracy. Besides that, Conversation Buffer with Summary Memory will enable efficient context retention of long-lasting interactions, making it a robust, user-centered platform for health and nutrition suggestions.

IV. EVALUATION

A. Evaluation Setup

To evaluate the Retrieval Augmented Generation (RAG) system, we conducted an evaluation that focused on factors such as relevance, faithfulness, performance, voice recognition, and memory retention. For relevance, the degree to which a response or piece of information is directly connected to and satisfies the user's query, the system is manually tested by asking questions related to meal plans, exercise routines, and general health advice, and the answers were rated on a 10-point scale based on how well they addressed the queries. The second metric, faithfulness, measures how well the system's responses align with provided resources, which is the PDF files in this case. We

evaluated how factually correct the system's answers were based on the retrieved documents, utilizing the RAG system itself to assess its responses. Different types of questions are used in the evaluation of RAG agent. There are four types of questions we applied to test the system, which are general, personalized, specific scenarios, and special considerations. The General questions contain general facts about nutrition and exercises, and Personalized questions are about recommendations for different types of people based on the given biometric data. For Specific Scenarios case, we asked the system to compare two different methods for a specific user's goal while we asked about recommendations for meals and workout plans for users that have special condition (e.g. some kind of disease) in Special Considerations case. After we obtained the ratings for each type of questions, the results for relevance and faithfulness are then classified into True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) based on the ratings. In addition to these classifications, we calculate Accuracy and F1 Scores to provide a more comprehensive evaluation of the system's metrics. Accuracy measures the overall correctness of the system by considering both correct and incorrect responses, while the F1 Score offers a balance between Precision (correctly identified relevant responses) and Recall (the ability to identify all relevant responses). These metrics are important as they give insights into how well the system is both identifying relevant and factually accurate responses, especially when dealing with complex or nuanced queries. The formulae for the Accuracy and F1 Scores are as follows:

$$Accuracy = \frac{TP + TN}{Total\ Responses}$$

$$F1\ Score = 2 \times \frac{Precision\ \times Recall}{Precision + Recall}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

The tests were performed on a typical computer, with users typing questions or interacting through a microphone to simulate a real-world scenario. For each response generated, we prompted the system to identify the source documents and evaluate how well its answers aligned with those sources. This self-assessment was compared against manual evaluations conducted by users. We prepared 10 example questions covering topics such as meal plans for specific diets, exercise routines for different fitness goals, and general health advice to clarify common myths. The self-evaluation process contains:

1. Asking the system to retrieve relevant excerpts from the documents that supported its answers.
2. Scoring its alignment with these sources on a 10-point scale, with higher scores indicating stronger consistency with the provided documents.
3. Comparing the system's self-assigned faithfulness scores to those determined manually by evaluators.

After that, performance was evaluated by measuring response time across 20 queries and calculating the average time taken to provide answers. Finally, we tested the system's chat memory by having users ask follow-up questions to determine if the system could remember previous interactions. Through this setup, we were able to evaluate how well the system performed in real-world conditions and identify areas for improvement.

B. Result and Discussion

We evaluated the performance of our lifestyle recommendation system across five key factors. The evaluation results for Relevance and Faithfulness are provided in the following table.

TABLE I
Evaluation Results for Relevance and Faithfulness

ID	Type	Relevance	Faithfulness
1	general	10/10 (TP)	10/10 (TP)
2	general	10/10 (TP)	9/10 (FP)
3	general	10/10 (TP)	10/10 (TP)
4	general	10/10 (TP)	10/10 (TP)
5	personalized	10/10 (TP)	9/10 (FP)
6	personalized	10/10 (TP)	9/10 (FP)
7	specific scenarios	9/10 (FP)	8/10 (FP)
8	specific scenarios	10/10 (TP)	10/10 (TP)
9	special considerations	10/10 (TP)	9/10 (FP)
10	special considerations	10/10 (TP)	9/10 (FP)

First, for relevance, the system achieved 80% accuracy. Out of 10 responses, 9 were classified as True Positives (TP), meaning the system accurately provided relevant answers. However, one response was classified as False Positives (FP), where relevance did not meet expectation. This resulted in an accuracy of 90% and an F1 score of 95%. The system demonstrated strong relevance score with clear and specific queries but occasionally struggled with queries requiring nuanced interpretation. To improve relevance, expanding the knowledge base with a broader range of high-quality documents would be beneficial. To improve relevance, expanding the knowledge base with a broader range of high-quality documents and implementing cross-encoder re-rankers can enhance the retrieval of relevant documents.

For faithfulness, we evaluated how factually correct the system's answers were based on the retrieved documents. The system's faithfulness achieved 40% accuracy, with 4 True Positives (TP) and 6 False Positives (FP). This resulted in an accuracy of 40% and an F1 score of 57%. Errors occurred when the system generated plausible but inaccurate answers. This issue, known as "hallucination," in AI, highlights the need to improve how the system retrieves and uses information. Solutions include fine-tuning embeddings with domain-specific datasets and leveraging hybrid retrieval techniques to enhance the accuracy of retrieved information.

In terms of performance, we measured the time it took to generate responses for both text and voice input. The average response time was 2 seconds for typed questions and 3 seconds for spoken questions due to additional processing for voice recognition. While the system was generally fast, it slowed down for longer, more complex queries. Optimizing the retrieval process using dynamic k adjustments and preprocessed embeddings can help speed up responses while maintaining accuracy.

For voice recognition, we tested the accuracy of converting spoken questions into text. The system achieved 85% accuracy in transcribing speech, performing well in quiet environments but struggling with noisy settings or heavy accents. Using Google's speech-to-text API was effective, but future improvements could focus on better handling accents and noise. Solutions include integrating a noise suppression library like PyAudioEffects and a voice activity detector (VAD) to enhance recognition accuracy.

Lastly, we evaluated chat memory, specifically the system's ability to remember previous interactions. It performed well 95% of the time, accurately recalling past questions and answers. However, it had some difficulty when switching topics during long conversations. While the current file-based memory system works well, transitioning to LangChain's Conversation Buffer Memory with a summarization feature can improve context retention and adaptability in complex, multi-topic interactions.

Overall, the lifestyle recommendation system performed well across most metrics. It provided relevant and accurate health advice and responded quickly to both typed and spoken queries. Areas for improvement include refining information retrieval to increase faithfulness, speeding up responses for complex questions, and improving voice recognition in noisy settings. Addressing these areas will help make the system more reliable and helpful for users seeking personalized dietary and wellness advice.

V. Conclusion

The development and evaluation of the lifestyle recommendation system utilizing Retrieval Augmented Generation (RAG) have demonstrated its potential as an innovative tool for personalized health care. With conversational memory, the system provides context-aware suggestions based on user needs, facilitating seamless interactions through voice or text. The RAG mechanism itself is a key innovation, which uses a database enhanced with embedded data from structured PDFs on fitness and nutrition to accurately respond to user queries. It uses advanced algorithms to retrieve relevant data, create insightful responses and integrate conversation history to provide a personalized experience. Its usability is further improved by adding features like text-to-speech synthesis, conversational memory and speech-to-text conversion, which enable users to access it in a variety of interaction modes. The system evaluation revealed that it performed well in terms of faithfulness and relevance, demonstrating its capacity to offer precise and significant suggestions for general health and fitness inquiries. Chat memory retention was reliable in 95% of the tested scenarios and response times were efficient averaging 2-3 seconds. However, there are areas that require improvement including mitigation AI "hallucinations" to ensure higher factual accuracy, optimization of retrieval mechanisms to process complex queries more quickly and the improvement of voice reignition accuracy in noisy environments or with different accents. In the future, upgrading the database with structured, biometric-based nutrition and fitness plans as well as more robust noise-handling and memory systems, will significantly enhance its utility. These advancements will ensure that RAG system develops into a dependable assistant for users looking for accurate, research-based lifestyle recommendations. This solution has the potential to completely transform how individuals manage their health and wellness with further development.

Acknowledgement

This research has been sponsored by Computer and Communication Engineering for Capacity Building Research Center, School of Applied Digital Technology, Mae Fah Luang University, Chiang Rai, Thailand.

References

[1] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," arXiv.org, Apr. 12, 2021.

[2] Y. Gao et al., "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv.org, Dec. 18, 2023.

[3] A. Bora and H. Cuayáhuitl, "Systematic Analysis of Retrieval-Augmented Generation-Based LLMs for Medical Chatbot Applications," Machine Learning and Knowledge Extraction, vol. 6, no. 4, pp. 2355–2374, Oct. 2024.

[4] Q. Guo, J. Tang, W. Sun, H. Tang, Y. Shang, and W. Wang, "SouLLMate: An Application Enhancing Diverse Mental Health Support with Adaptive LLMs, Prompt Engineering, and RAG Techniques," arXiv.org, 2024.

[5] Richard-Ojo, O., Wimmer, H., & Rebman Jr, C. M. (2024). RAG Chatbot for Healthcare Related Prompts Using Amazon Bedrock. Proceedings of the ISCAP Conference. Georgia Southern University.

[6] Y. Haj, "RAG programming on LLMs to improve the quality of life of elderly people," Riunet.upv.es, Oct. 2024.

[7] Y. J. Oh, J. Zhang, M.-L. Fang, and Y. Fukuoka, "A systematic review of artificial intelligence chatbots for promoting physical activity, healthy diet, and weight loss," International Journal of Behavioral Nutrition and Physical Activity, vol. 18, no. 1, Dec. 2021.

[8] Hou, Y., & Zhang, R. (2024). Enhancing dietary supplement question answering via Retrieval-Augmented Generation (RAG) with LLM.

[9] Yang, Z., Azimi, I., Zaki, M. J., Gaur, M., Seneviratne, O., McGuinness, D. L., Rashid, S. M., & Rahmani, A. M. (2024). Transforming Personal Health AI: Integrating Knowledge and Causal Graphs with Large Language Models. Proceedings of the ISCAP Conference.