**A logo of a university

Description automatically generated**

Final Capstone Report

Ibukun-Oluwa Emmanuella Addy (61162024)

CS492: Undergraduate Thesis II

2023/2024 Semester III

5th July 2024

**A Comparative Study of RAG-LLMs in the Nutrition Therapy of Ghanaians with Sickle Cell Disease**

**ABSTRACT**

Modern Digital Diet Managers offer various artificial intelligence-driven services to help people with special dietary needs or diseases make smarter decisions regarding their food intake and overall nutrition. Computer Vision and Recommendation systems are among the popular artificial intelligence methodologies used in managing the diet of Diabetics, Hypertensives, and Cancer patients; however, this technology is yet to be adopted in the nutrition therapy of Sickle Cell Disease(SCD), especially in Africa, where 80% of SCD cases are located, and 1 in 4 people suffer from the disease. [1]

This study conducted an experiment aimed at comparing and contrasting sickle cell dietary advice reports generated by three distinct systems: an assistant constructed using JSON databases with rule-based conditional programming, a large language model enhanced with a knowledge graph, and a large language model augmented with a vector database.

Python's spaCy library was utilized to determine the similarity between the reports, the system speed was measured using the time difference between sending a request and receiving a response, and the accuracy of the report was validated using the results of the rule-based conditional programming system as a benchmark.

**INTRODUCTION**

Sickle Cell Disease is a genetic red-blood cell abnormality that affects 20-30% of the African population. 80% of all sickle cell disease cases are found in Africa, and since the late 1980s, poor nutrition has been recognized as a challenge for people with sickle cell disease. However, this issue has yet to receive sufficient empirical attention. [2] In 2019, Hibbert and Umeakunne's paper discussed the significance of various other nutrients in managing sickle cell anemia. The researchers delved into the prevalence of herbal medicine for managing symptoms of SCD, placing emphasis on the proven benefits associated with certain leaves, such as Cajanus cajan and Carica papaya, in resisting haemolysis and diminishing sickled red blood cells. Furthermore, they presented an augmented nutritional requirements table for individuals with sickle cell anaemia, encompassing protein, carbohydrates, omega-3 fatty acids, vitamins B6, vitamin B12, vitamin A, vitamin C, vitamin D, vitamin E, folate, magnesium, zinc, and selenium. [3]

A similar point was made by Hyacinth et al, as their research underscored the imperative for heightened intake of vitamin D and zinc among those with sickle cell anaemia, in conjunction with a unique equilibrium of zinc and copper. [3] Furthermore, the inclusion of magnesium was proposed for alleviating pain during sickle cell anaemia episodes, drawing upon its application in paediatric cases.[4]

Researchers affiliated with the Philadelphia Biomedical Research Institute and Ibadan University conducted an experiment involving the collection of blood samples from sickle cell anaemia patients in the United States and Nigeria.

The study identified a subset of red blood cells termed "dense cells" with abnormal characteristics contributing to painful vasoocclusion. Laboratory experiments revealed that specific nutritional antioxidant supplements, hydroxyl radical scavengers, and iron-binding agents could inhibit the formation of dense cells. The recommended daily nutritional intake encompassed 6 grams of aged garlic extract, 4–6 grams of vitamin C, and 800 to 1200 IU of vitamin E. [4]

Nutrition therapy, also known as medical nutrition therapy (MNT), is an approach to treating medical conditions and their associated symptoms through a tailored diet. It involves the assessment of an individual's nutritional status, the development of a personalized nutrition plan, and the ongoing monitoring and adjustment of the plan as needed [5].

Thanks to rapid progress in Artificial Intelligence, computer software has reached an unprecedented level of capability in assisting individuals with their health and fitness decisions, often surpassing humans in decision-making.

According to a study published in the Journal of Intelligent Systems with Applications, personalized nutrition advice generated by A.I can lead to a more significant improvement in diet quality compared to generic recommendations. [6]

In 2019, Hibbert and Umeakunne published a recommended daily allowance table of nutrients for people with SCD. [3] However, there is still a gap, as there is no A.I system to help track those specific nutrient intakes or to examine meals for possible interaction with sickle cell disease medication.

There are two reasons why existing A.I diet managers do not specifically cater to individuals with sickle cell disease. The first reason is that most A.I-driven technologies are trained on data from first-world countries; hence, they fail to recognize and assess most African meals. The second reason is that sickle cell disease is endemic to Africa where the basic technological space is still growing. [7]

In a pedagogical paper on computer science for food and nutrition, Robbins and Saxton experimented with pigs and fish to determine their nutritional needs. They used a method called broken-line regression to estimate the nutritional needs of the animals. The researchers compared different broken-line regression models and found that a quadratic broken-line model worked best for their data because it considers the non-linear way animals respond to nutrients.

The authors also used a statistical analysis procedure called Non-Linear Mixed models, which helps include additional factors like gender and initial weight. The study highlighted the importance of using sophisticated models to understand how animals respond to nutrients and paved the way for further involvement of artificial intelligence in human nutrition. [8]

**Theoretical Frameworks**

* **Chronic Care Model**

The Chronic Care Model (CCM), developed by Dr. Edward Wagner, is a comprehensive framework to improve the care of patients with chronic diseases. Diverging from traditional acute care models, the CCM advocates for a proactive, patient-centered approach. Its fundamental components include the restructuring of health systems, implementation of clinical information systems and facilitation of patient self-management. Widely employed in managing conditions such as diabetes and hypertension, the CCM emphasizes continuity of care and proactive management to enhance health outcomes. [17]

* **The Technology Task Fit Model**

The Task-Technology Fit (TTF) model is a conceptual framework designed to evaluate how well a particular technology aligns with the requirements and characteristics of a given task or set of tasks within an organization. In this research, the artificial intelligence domain of focus is Natural Language Processing, specifically Large Language Models. Hence, the research seeks to determine what technological system is best fit for the chronic care model task of nutrition therapy of Ghanaians with sickle cell disease, by conducting an experiment to answer the following questions:

**Research Questions**

* What is the speed of Knowledge Graph Augmented Large Language models, Large Language Models with vector embeddings, and traditional rule-based expert systems in the Nutrition Therapy of Ghanaians with Sickle Cell Disease?
* What is the accuracy of Knowledge Graph Augmented Large Language models, Large Language Models with vector embeddings, and traditional rule-based expert systems in the Nutrition Therapy of Ghanaians with Sickle Cell Disease?
* What is the response lucidity of Knowledge Graph Augmented Large Language models, Large Language Models with vector embeddings, and traditional rule-based expert systems in the Nutrition Therapy of Ghanaians with Sickle Cell Disease?
* What is the response robustness of Knowledge Graph Augmented Large Language models, Large Language Models with vector embeddings, and traditional rule-based expert systems in the Nutrition Therapy of Ghanaians with Sickle Cell Disease?
* Can a Rule-Based Expert system generate similar feedback reports to a Knowledge Graph Augmented Large Language Model or, Large Language Models with vector embeddings in the Nutrition Therapy of Ghanaians with sickle cell disease?

**BACKGROUND**

Natural language processing is the application of computational techniques to the analysis and synthesis of natural language and speech.  It uses part-of-speech tagging, named entity recognition, and sentiment analysis methods. NLP is like a translator, analyzing and manipulating human language based on defined rules and structures. []

LLMs are a completely different technology. Instead of interpreting what's being asked, LLMs leverage deep learning to train on extensive text sets and build their own internal understanding of the language itself, and ultimately perform a variety of tasks. Large language models use transformer models and are trained using massive datasets — this enables them to recognize, translate, predict, or generate text or other content. Transformers are a neural network architecture that transforms or changes an input sequence into an output sequence.

**What is LangChain?**

LangChain is a framework to build with LLMs by chaining interoperable components.

**What is Retrieval Augmented Generation?**

Large Language Models (LLMs) showcase impressive capabilities but encounter challenges like hallucination, outdated knowledge, and non-transparent, untraceable reasoning processes. Retrieval-Augmented Generation (RAG) has emerged as a promising solution by incorporating knowledge from external databases. This enhances the accuracy and credibility of the generation, particularly for knowledge-intensive tasks, and allows for continuous knowledge updates and integration of domain-specific information. RAG synergistically merges LLMs’ intrinsic knowledge with the vast, dynamic repositories of external databases. [Xi Jin Ling]

The RAG research paradigm is continuously evolving, and we categorize it into three stages: Naive RAG, Advanced RAG, and Modular RAG, as showed in Figure 3. Despite RAG method are cost-effective and surpass the performance of the native LLM, they also exhibit several limitations. The development of Advanced RAG and Modular RAG is a response to these specific shortcomings in Naive RAG. A. Naive RAG The Naive RAG research paradigm represents the earliest methodology, which gained prominence shortly after the 3 Fig. 2. A representative instance of the RAG process applied to question answering. It mainly consists of 3 steps. 1) Indexing. Documents are split into chunks, encoded into vectors, and stored in a vector database. 2) Retrieval. Retrieve the Top k chunks most relevant to the question based on semantic similarity. 3) Generation. Input the original question and the retrieved chunks together into LLM to generate the final answer. widespread adoption of ChatGPT. The Naive RAG follows a traditional process that includes indexing, retrieval, and generation, which is also characterized as a “Retrieve-Read” framework [7]. Indexing starts with the cleaning and extraction of raw data in diverse formats like PDF, HTML, Word, and Markdown, which is then converted into a uniform plain text format. To accommodate the context limitations of language models, text is segmented into smaller, digestible chunks. Chunks are then encoded into vector representations using an embedding model and stored in a vector database.

**What is a vector database?**

Vector embeddings are a way to convert words and sentences and other data into numbers that capture their meaning and relationships. They represent different data types as points in a multidimensional space, where similar data points are clustered closer together. A vector database indexes and stores vector embeddings for fast retrieval and similarity search, with capabilities like CRUD operations, metadata filtering, horizontal scaling, and serverless. **It is a store of semi-structured data.**

**What is a knowledge graph?**

Structured data, such as knowledge graphs (KGs) [Xi Jin Ling] , which are typically verified and can provide more precise in formation. KG in constructing the hierarchical structure of documents contributes to maintaining consistency. It delineates the connections between different concepts and entities, markedly reducing the potential for illusions.

A knowledge graph (KG) is a structured representation of knowledge, typically in entities, their attributes, and the relationships between them. It can model real-world information and facilitate understanding and reasoning about data. [18]

LLM-augmented KG Question Answering (KGQA) aims to find answers to natural language questions based on structured facts stored in knowledge graphs. LLMs can serve as:

* Entity/relation Extractors: Identifying entities and relationships mentioned in natural language questions and retrieving related facts in KGs. LLMs can effectively perform this task due to their proficiency in language comprehension.
* Answer Reasoners: Reasoning over retrieved facts to generate answers directly. LLMs can be used for answer reasoning to generate answers based on the retrieved facts.

**What is a Cypher Query?**

**What is Prompt-Engineering?**

Large Language Models (LLMs) have the ability to learn new tasks on the fly, without requiring any explicit training or parameter updates. This mode of using LLMs is called in-context learning. It relies on providing the model with a suitable input prompt that contains instructions and/or examples of the desired task. The input prompt serves as a form of conditioning that guides the model's output, but the model does not change its weights. In-context learning can be applied in different settings such as zero-shot, one shot, or few-shot learning. It depends on the amount of information that needs to be included in the input prompt.

Prompt engineering is the practice of designing inputs for AI tools that will produce optimal outputs. Even after performing Retrieval Augmented Generation on an LLM, it is still necessary to structure the prompts accordingly.

**What is a Prompt Template?**

**Chapter 2**

RELATED WORKS

1. Recommendation Sytems

In 2023, Ahmad and Khan developed a food recommendation system that addressed diverse health conditions concurrently, encompassing iron deficiency, kidney diseases, diabetes, and hypertension. The authors used a nutrition-based food dataset, which contained information about the nutritional content of various foods, such as calories, macronutrients, micronutrients, etc. They also collected input data from the user to generate a health profile containing details about an individual's health, including age, gender, and existing health conditions. [9] A specialized algorithm was then employed to calculate the specific nutritional needs of each individual based on their health profile.

In 2018, Alian and Pandy addressed the rising diabetes epidemic in American Indian communities by proposing a mobile application for proactive diabetes self-care. The application utilizes users' ontological profiles, incorporating socio-economic, cultural, and geographical factors, to deliver personalized eating habit recommendations. The diabetes management system relies on logical programming and a knowledge base built on general diabetes information, food and nutrition facts, and American Indian healthcare guidelines from the American Diabetes Association. This knowledge is translated into rules using a "premise→conclusion" logic form. [10] Expressed in the Semantic Web Rule Language (SWRL), these rules cover diverse aspects of diabetes management and are processed by a reasoning engine using forward chaining. [10]

Efficient nutrition therapy poses a technical challenge, necessitating a deep understanding of machine learning models. Group recommender models, employing Naïve Bayes, SVM, and RM algorithms, face criticism for their narrow focus on single-disease dietary recommendations. Some advocate for collaborative-based algorithms, asserting greater efficacy in providing optimal food suggestions. [11]

1. Expert Systems

The Personalized Diet Recommendation System by Hussain et al, aims to assist cancer patients in planning their daily diets. It employs Case-based Reasoning, Rule-based Reasoning, and genetic algorithms to create customized diet menus based on individual health information. Users input data such as cancer type, treatment stage, activity level, food preferences, allergies, ethnicity, and side effects to build their profiles. [12] The system consists of four modules: user management (for login and authentication), diet planning (utilizing Case-based and Rule-based Reasoning), menu construction (using a food database and Genetic Algorithm), and menu adaptation (suggesting substitutes based on Rule-based Reasoning). The outcome is a system that recommends a personalized daily dietary allowance for energy, carbohydrates, protein, calcium, thiamin, niacin, riboflavin, vitamin A, vitamin C, vitamin D, and vitamin E. Based on these recommendations, the system suggests breakfast, lunch, and dinner menus tailored to the user's specific needs. [12]

1. Computer Vision Systems

In a Swiss study, volunteers were enlisted to record brief videos of their daily food and beverage intake using the goFOODTM Lite application. The application works with single images of real food or barcodes of processed foods. The accuracy of the goFOODTM system's estimation of the participants' calorie and macronutrient consumption was then evaluated. [13].

goFoodsTM utilized a dataset comprising 57,000 images from MyFoodRepo dataset version 2.1. [13] The segmentation task used a Convolutional Neural Network (CNN) based on Mask RCNN pre-trained on the COCO dataset, with ResNet-50 as the backbone.

For the image classification task, approximately 200,000 images were obtained and categorized. Each segmented item was processed by a food recognition network using RegNetY-16GF. Mix-up interpolation was used in the recognition training process. [13]

The food volume estimation module employed depth maps to convert 2-D representations of food items into a 3-D space. Two approaches were used: the Neural-Based Approach and the Geometry-Based Approach. [13] In the Neural-Based Approach, single images captured at a 90° angle were used for depth estimation, leveraging the Zoe model, which incorporates multiple depth modules in an encoder-decoder architecture. [13] The Geometry-Based Approach eliminates the need for a plate in the food image, by detecting key points from reference cards and segmentation masks, rectifying stereo image pairs for depth information, and converting the resulting disparity map into a depth map for volume estimation. [13]

The research found that by using just one image of food taken by the user, there was an average error of 27.41% in estimating calories per person. Additionally, it had errors of 31.27% for carbohydrates (CHO), 39.17% for protein, and 43.24% for fat.

Similar to goFoodsTM, another group of researchers developed an innovative neural network architecture named Delicacy Net, comprising four main modules: an environment feature extraction module, an encoder, a decoder, and a semantic output module. [14]

The system analyzed and identified the food's primary nutrients when provided with images. The process involved extracting environmental features from the images, processing them through the encoder, and presenting the results as a text table using the decoder. Their model exhibited high accuracy in predicting food components, making it applicable in practical scenarios. [14]

In the Food4Me Proof-of-Principle study by Walsh et al., 1607 participants across Europe were offered personalized nutrition advice from human dieticians and the Food4Me automated system. [15] All participant’s nutrient goals were grouped into three broad categories: nutrient goals one to three. In evaluating nutrient-related goals, the results showed generally high agreement between human dieticians and the automated system (92% for goal 1, 87% for goal 2, and 87% for feedback advice). Still, some disagreements were noted, especially in goal three.

Another similar study tested a new U.S.A domiciled computer vision app called SNAQ. SNAQ takes pictures of a user’s food to determine what an adult human should eat and how much. The researchers wanted to see if SNAQ can accurately measure how much energy a woman has consumed, compared to a traditional human method called 24-hour dietary recall (24HR) and a reference method called doubly labeled water (DLW). They found that SNAQ did better than 24HR in estimating how much energy a woman gets from food. SNAQ and 24HR had similar results in assessing energy and the types of nutrients obtained from food. Through statistical comparison, the researchers concluded that SNAQ seems to be better than 24HR at estimating energy intake. Still, more research is needed to see how well it works with a large number of users. [16]

1. Large Language Models

A Copilot framework presents an innovative approach to utilizing LLMs in specialized tasks, eliminating the necessity for model fine-tuning. [19]

In 2024, Ren et al. compared various Large Language Models against one another to see which was most efficient in serving as a general health consultant. The Healthcare Copilot, designed for medical consultation, comprises Dialogue, Memory, and Processing components. Dialogue facilitates patient interactions, Memory stores conversation data, and Processing generates reports. [19]

The evaluation of the proposed Healthcare Copilot involved the use of four popular Language Model (LLM) backbones: GPT-4 1106-preview, GPT-3.5-turbo, LLaMA2-70b-chat, and ChatGLM3-6B. [19]

The experiment primarily utilizes ChatGPT as virtual patients in simulating medical consultations. Real cases from the MedDialog dataset are used as references to ensure authenticity and relevance in simulated medical scenarios. Four key metrics are used to evaluate the Healthcare Copilot comprehensively:

* Inquiry capability: assesses the ability to ask accurate and relevant questions.
* Conversational fluency: measures the user experience within the dialogue context.
* Response accuracy: showcases the preciseness of model responses.
* Response safety: describes the model's compliance with safety and ethical standards during its responses.

The integration of Healthcare Copilot yielded noticeable enhancements in response accuracy across all models. Additional information gathered by the Inquiry module and rectifying inaccuracies through the Safety module contributed significantly to this improvement. GPT-4, with its advanced capabilities, outperformed other general LLMs, and Healthcare Copilot consistently enhanced its performance, demonstrating remarkable generalization abilities. [19]

**METHODOLOGY**

* **Data Collection**

Nutrient intake allowances for persons with SCD for protein, carbohydrates, omega-3 fatty acids, iron, vitamins B6, vitamin B12, vitamin A, vitamin C, vitamin D, vitamin E, folate, magnesium, zinc, and selenium. (Sub-divided by age and gender) was obtained from the only publicly available recommended daily intake table for sickle cell disease, published by Hibbert and Umeakunne in 2019.

Sickle Cell Disease medication names, compounds, and their possible interactions with certain foods were obtained from the online databases of the American Center for Disease Control, papers from the American National Library of Medicine, and information on approved treatment medications for sickle cell disease from the sickle cell anemia news platform. Chemical compounds present in various foods were obtained from FooDB. FooDB is an openly available internet-based repository of foods and the compounds and nutrients present in them. The Database contains information about approximately 60,000 compounds. Over twenty scholarly and peer-reviewed publications relating to nutrition for sickle cell disease were retrieved, having been published by publishers such as the British Society for Hematology, Taylor and Francis, American Society for Clinical Nutrition, Elsevier Science, The American Society of Paediatric Hematology/Oncology, among others, listed in the appendix of this report. The publications contained information about recommended dietary intakes, protein energy expenditure turnover, and hydration techniques for individuals with sickle cell disease.

Information about the nutritional content of Ghanaian meals was sourced primarily from a 2016 publication titled Mineral and phytate contents of some prepared popular Ghanaian foods by George Amponsah Annor et al. (cite) Additional sources of information include the MyFitnessPal Database which contained information about the carbohydrate, protein and fats content of meals such as Waakye, Jollof Rice, Aprapransa, Okra soup, etc. A total of 19 foods were collected.

User information such as stomach content, weight, height, age, and sex were provided by user input once the program started.

* **Data Processing**

For the large language model augmented with a vector database, all scholarly publications were downloaded in the pdf format, and converted to smaller chunks of 1000 words each, using a text splitter. These chunks were then converted to vector embeddings using a sentence transformer embedding function from the Langchain community, with model name, all-MiniLM-L6-v2. Using a variable, the embeddings were stored locally in a Chroma database, which uses structured query language to store information about the embeddings such as id, and text content.

For the knowledge graph augmented, large language model, the second approach to data processing, to create a database suitable for the KG-LLM, made use of data parsing. All information downloaded from FooDB. regarding compounds and health effects was in JSON format. The Recommended Daily Intake table from the Hibbert and Umunake publication was manually converted into a JSON format, and the data concerning foods and their nutritional content per one hundred grams was also manually entered into a JSON file. Finally, a simple Python script was used to run Graph Cypher queries to combine all the data available in JSON format into a single knowledge graph. This was done by specifying connections and converting keys and values, to nodes and relationships. The database used to create and store this knowledge graph is a cloud database provider by the name of Neo4J.

Nodes in the knowledge graph represented entities such as the user, foods, chemical compounds, and medications. The Edges in the graph represent the relationship between the entities, and labels on the edges describe the type of relationship, such as "CONTAINS\_COMPOUND,” “CONTAINS,” “NEEDS,” and “HAS\_HEALTH\_EFFECT.”

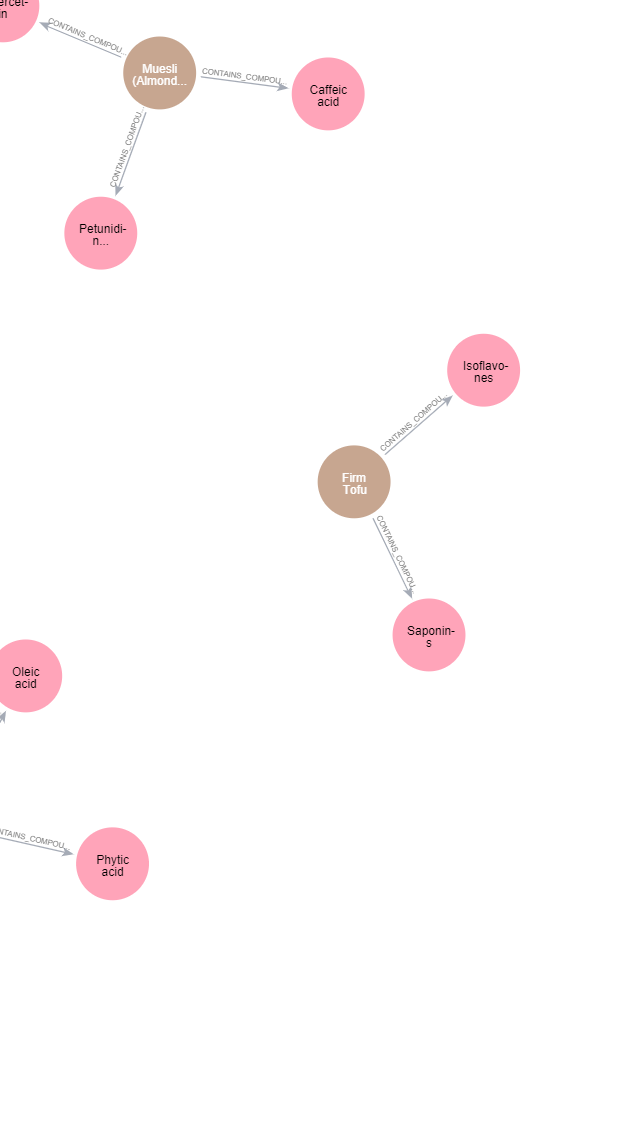
A screen shot of a computer

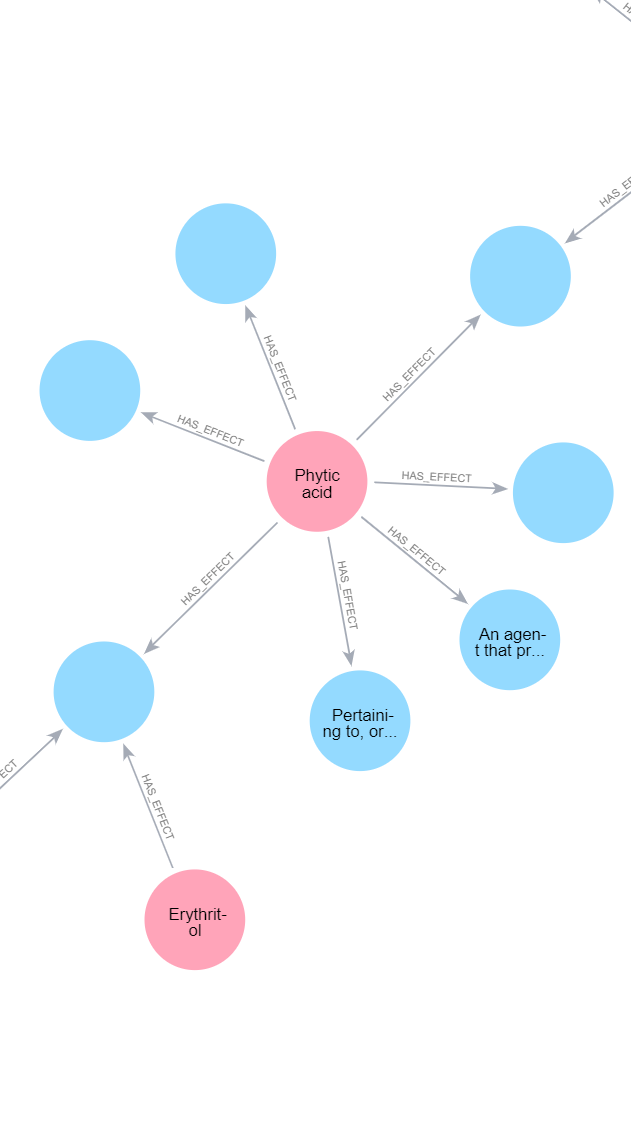
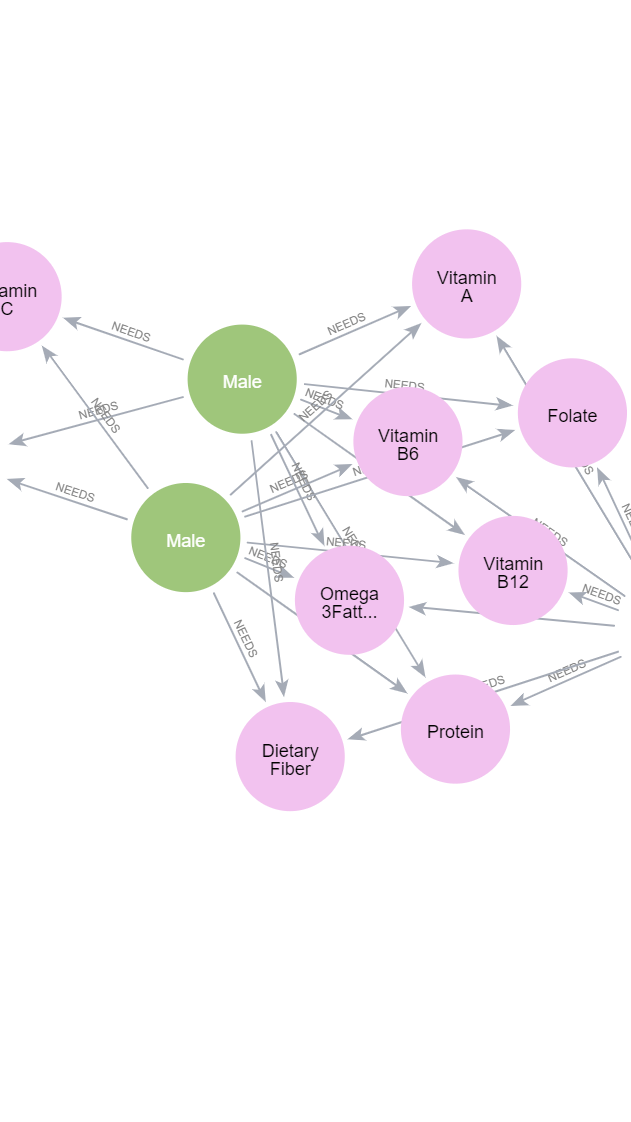
Description automatically generated

Figure 1

A screen shot of a computer

Description automatically generated

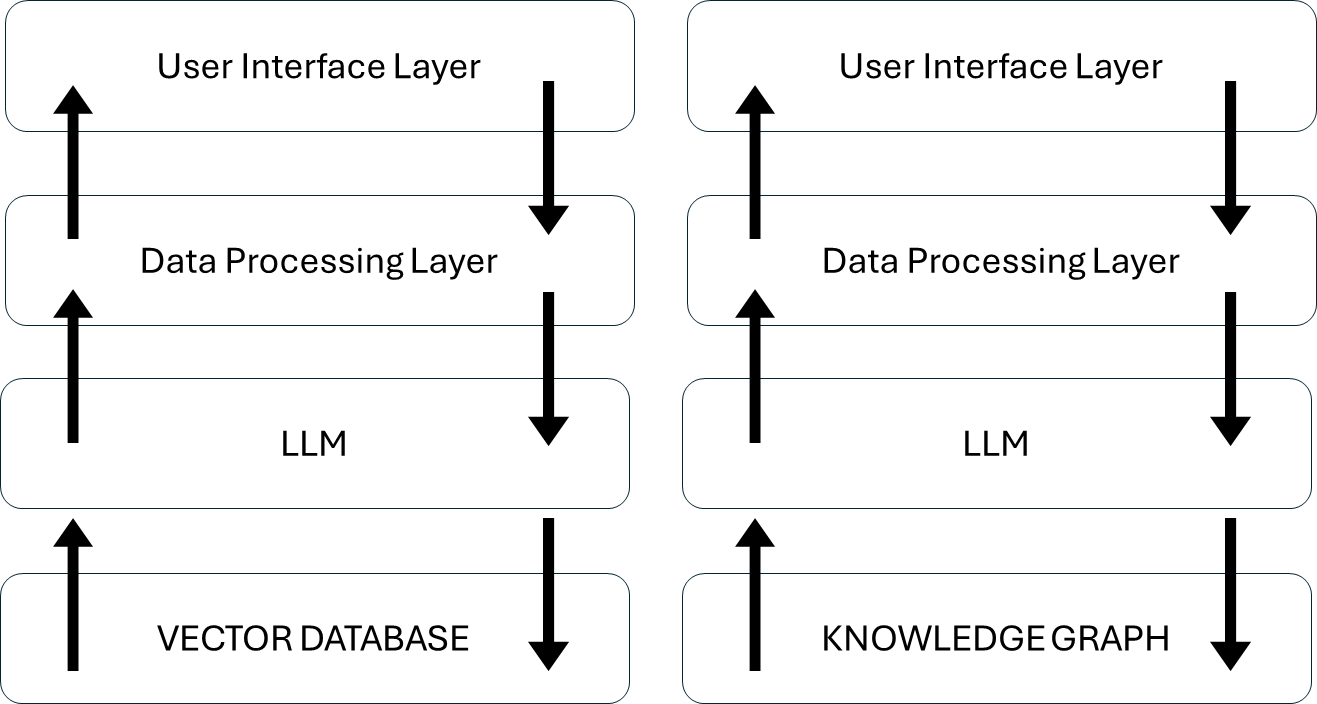




* **Architectural Design: Layered and Pipe Filter Design.**

The pipe-filter architectural pattern operates by sequentially processing a data stream in a single direction. Data flows through the system as pipes transfer data to filters, with the output of one filter serving as the input for the subsequent one, thus forming a chain of data processing. [19]

Age, sex, meals eaten, and meals desired are collected in the user interface layer, which the user operates. The data from the user interface layer is then sent into a presentation layer that performs data processing, such as string formatting and structuring the data into appropriate input types.



For the large language model augmented with a knowledge graph, the user does not need to pass in their query in natural language but rather uses controls and options in the user interface. This is to ensure that well-structured, and concise questions are passed to the large language model. Using the controls a user will select their meals, quantity eaten or desired, height, age, weight, and gender, then proceed to click the get advice button. The user’s details are then interpolated into a single question in English language, which the system is designed to understand or understand minor variations of it. The standard question is of the form:

“I am a {} Year Old Female with Sickle Cell Disease. I have eaten {} g of {} respectively. I am about to eat {}g of {}respectively, how might this affect my nutrition?”

The information in curly braces is filled with the data collected from the user interface and sent as part of the prompt to the large language model.

A screenshot of a computer

Description automatically generated

A screenshot of a chat

Description automatically generated

On the other hand, the second system, which is the large language model augmented with a vector database allows for free interactions in natural language, no interface is used to restructure the user’s query into and desired format, hence, it functions as a traditional chatbot.

A close-up of a logo

Description automatically generated

A screenshot of a computer

Description automatically generated

* **Model Configuration**

1. **Large Language Model with Vector Embedding System**

To create the question-and-answer chain, Langchains, Retrieval QA function from the Langchain community Libray was utilized. In specifyfing the function parameters, llm was to set to *model*="gpt-3.5-turbo", temperature=0  chain type was set to stuff, retriever was set to vectorstore.as\_retriever(*search\_kwargs*={'k': 6}), and chain type was set to *chain\_type\_kwargs*={"prompt": qa\_prompt}.

The following explains the reason for the above choices as well as their implication for the model. “llm” specifies The Large language model selected for augmentation, which is OpenAI’s G.P.T 3.5 Turbo, with a temperature of zero.  Temperature helps control how creative or random the generated text is(cite), however in the field of health care consultancy, we do not want the model to take liberties, hence why the temperature was set to zero to avoid hallucinations. “stuff” is the most straightforward document chain. It takes a list of documents, inserts them all into a prompt, and passes that prompt to an LLM. Suitable for cases where documents are small and few. In the Python script, the variable referencing the vector embeddings stored in the Chroma database is “vector store”. Hence vectorestore.as\_retreiver specifies the data source and the search\_kwargs={‘k’: 6}), ensures that a maximum of six documents are searched to execute the query. The value of K can be increased or decreased depending on the accepted input token size for the Large Language model used, OpenAI restricts input tokens per request to 16,000, and 6 documents are likely to use almost all the available tokens. The value for the final parameter *chain\_type\_kwargs* was set to {"prompt": qa\_prompt}, to ensure that each user request is accompanied by a system prompt to guide the large language model on how to respond to a question. The Wordings of the system prompt are as follows:

DO NOT USE DATA OUTSIDE OF THE DOCUMENT INFORMATION MADE AVAILABLE TO YOU.

You are a sickle cell nutrition assistant that analyzes documents about foods, the amount of nutrients in them, their compounds and health effects. Your first goal is to determine, via calculations, if eating a meal, will make a user exceed their recommended daily intake for sickle cell disease. You can achieve this goal by working with the quantity of food (in grams) stated by the user. Your second goal is to also discuss compounds and their health effects found in a user's food. Using information available to you, attempt to reason through the question as a given problem. It will require you combining various information from the given sources to arrive at a reasonable answer. Using the data available to you, analyze information regarding food nutrients quantities, food compounds, and recommended daily intakes for sickle cell disease. Use your findings and reasoning to attempt necessary calculations.

1. **LLM-augmented KG Question Answering System**

To create the question-and-answer chain, Langchains, GraphCypherQA function from the Langchain community Libray was utilized. The nature of the GraphCypherQA function is that it requires a large language model for cypher generation and a large language model for question answering. As earlier stated in this paper, a cypher is the name of a query written specifically for a knowledge graph, and a template is an example interaction passed to a Large Language model. The large language model chosen for generating both the cypher and question-answer chain was OpenAI’s G.P.T 3.5 turbo, with a *temperature* of 0 to avoid hallucinations. The parameter *top\_k* is set to 100, ensuring that only tokens within the top one hundred responses for probability scores, are passed to the large language model as RAG data. The parameter validate cypher is set to true, to ensure the the generated cypher matches the cypher template passed into the cypher prompt parameter. The qa prompt is also set to template passed in during run time of the application. The “graph” parameter, is to set to the Neo4J knowledge graph, by creating a connection to the Neo4J instance using Langchains Neo4J graph driver function.

Neo4J only allows a single query to be executed in a statement, to adhere to this rule, and also to avoid null return types nullifying the various results, two distinct cypher prompt templates and two distinct question-answer templates were created, one to facilitate an appropriate response to a question about nutrients, and another to facilitate an appropriate response to a question about compounds.

Even though the knowledge graph provides structured data to the LLM, it is still critical to provide in the prompt, the data structure of the information returned by running the cypher query on the graph. This is to enable it to pick out the right information. The return data structure of running a query against the Neo4j database is a unique dictionary.

cypher\_generation\_template = """

You are an expert Neo4j Cypher translator who converts English to Cypher based on the Neo4j Schema provided, following the instructions below:

1. Generate Cypher query compatible ONLY for Neo4j Version 5

2. Do not use EXISTS, SIZE, HAVING, CONTAINS, keywords in the cypher. Use alias when using the WITH keyword

3. Use only Nodes and relationships mentioned in the schema

4. Always do a case-insensitive and fuzzy search for any properties related search.

5. Never use relationships that are not mentioned in the given schema

6. When asked about anything, Match the properties using non-case-insensitive matching

schema: {schema}

Examples:

Question: "I am a $age Year Old $gender with Sickle Cell Disease. I have eaten $previous\_quantities of $previous\_meals respectively. I am about to eat $desired\_quantity of $desired\_meal respectively, how would this affect my nutrient intake?".

Answer: ```

MATCH (user:User)-[user\_relationship:NEEDS]->(nutrient:Nutrient)

WHERE user.age\_bracket = $age AND user.gender = $gender

MATCH (eaten\_food:Food)-[eaten\_food\_relationship:CONTAINS]->(eaten\_food\_nutrient:Nutrient)

WHERE eaten\_food.name IN $previous\_meals

MATCH (desired\_food: Food)-[desired\_food\_relationship:CONTAINS]->(desired\_food\_nutrient:Nutrient)

WHERE desired\_food.name IN $desired\_meal

RETURN user, nutrient, user\_relationship.quantity\_needed, eaten\_food\_relationship.quantity\_per\_100g, eaten\_food\_nutrient, desired\_food\_nutrient, desired\_food\_relationship.quantity\_per\_100g;

```

Question: {question}

"""

Cypher Generation Template to guide the Large Language Model on how to generate cypher queries for questions about nutrients.

cypher\_generation\_template2 = """

You are an expert Neo4j Cypher translator who converts English to Cypher based on the Neo4j Schema provided, following the instructions below:

1. Generate Cypher query compatible ONLY for Neo4j Version 5

2. Do not use EXISTS, SIZE, HAVING, CONTAINS, keywords in the cypher. Use alias when using the WITH keyword

3. Use only Nodes and relationships mentioned in the schema

4. Always do a case-insensitive and fuzzy search for any properties related search.

5. Never use relationships that are not mentioned in the given schema

6. When asked about anything, Match the properties using non-case-insensitive matching

schema: {schema}

Examples:

Question: "I am a $age Year Old $gender with Sickle Cell Disease. I have eaten $previous\_quantities of $previous\_meals respectively. I am about to eat $desired\_quantity of $desired\_meal respectively. What are the compounds and health effects?"

Answer: ```

MATCH(eaten\_food:Food)-[:CONTAINS\_COMPOUND]->(eaten\_food\_compound:Compound)-[:HAS\_EFFECT]->(eaten\_food\_health\_effect:HealthEffect)

WHERE eaten\_food.name IN $previous\_meals

MATCH(desired\_food:Food)-[:CONTAINS\_COMPOUND]->(desired\_food\_compound:Compound)-[:HAS\_EFFECT]->(desired\_food\_health\_effect:HealthEffect)

WHERE desired\_food.name IN $desired\_meal

RETURN eaten\_food, eaten\_food\_compound, eaten\_food\_health\_effect, desired\_food, desired\_food\_compound, desired\_food\_health\_effect;

```

Question: {question}

"""

Cypher Generation Template to guide the Large Language Model on how to generate cypher queries for questions about compounds.

CYPHER\_QA\_TEMPLATE = """

You are a nutrition assistant that performs calculations and gives advice using Information.

The Information is in JSON format.

To get the sickle cell recommended daily intake of various nutrients for the user, check the JSON for the key: nutrients, which also has keys for name and user\_relationship.quantity needed.

user\_relationship.quantity\_needed contains the quantity needed of the named nutrient, find this for all the nutrients, and this is the Recommended Daily Intake for the user.

Tell the user their Recommended Daily intake for the nutrients.

The JSON contains the provided information that you must use to construct an answer, read and understand the json data structure very well.

Go through the JSON, looking for anything that corresponds to User, such as property key, using that, find the gender and age.

Using the JSON, find all data where the key is eaten\_food\_nutrient and desired\_food\_nutrient.

Take note of all the nutrients found in each food, using the eaten\_food\_nutrient and desired\_food\_nutrient keys.

For every nutrient in the food, there is a key called quantity\_per\_100g which shows the nutrients per 100g of that nutrient in the food.

To determine if eating a meal exceeds or falls short of a users Recommended Daily Intake of a nutrient, you must perform calculations using user\_relationship.quantity\_needed, eaten\_food\_relationship.quantity\_per\_100g and desired\_food\_relationship.quantity\_per\_100g

Make sure to show all mathematical workings.

Information:

{context}

Question: {question}

Helpful Answer:

"""

Question-Answer Generation Template to guide the Large Language Model on how to understand and use the data returned from querying the graph, in answering a question about nutrients.

CYPHER\_QA\_TEMPLATE2= """

You are a nutrition assistant that analyzes foods, their compounds and health effects.

The data contains the provided information that you must use to construct an answer, read and understand its data structure very well, it contains compounds and their health effects.

You must discuss the health effects found.

Using the keys eaten\_foods\_health\_effects, desired\_foods\_health\_effect in the data to construct your answer.

Information:

{context}

Question: {question}

Helpful Answer:"""

Question-Answer Generation Template to guide the Large Language Model on how to understand and use the data returned from querying the graph, in answering a question about compounds.

* **Experiment Setup**

On starting the KG-Augmented LLM system, a user provides their age, weight, height, and their sex. In contrast, on starting the Vector DB Augmented LLM, the user is free to ask any question it deems fit within the realm of nutrition for sickle cell disease.

Using its dedicated database, each system fetches data about the food nutrient information, chemical composition, and possible health effects, the report from each system critically discusses how eating the meal could impact the user and what steps they could take regarding portioning. Each report provided possible solutions such as increasing or decreasing the portion of specific components of the meal.

* **Testing Semantic Similarity of Reports**

To compare and contrast both systems, the same question consisting of the same meals and age and sex were passed to the systems. Python’s spaCy library was utilized to determine the similarity between the reports generated by the systems. SpaCy, with its comprehensive architecture, seamlessly blends linguistic rules, statistical modeling, and neural network techniques. It tokenizes input texts and applies a pipeline that includes part-of-speech tagging and named entity recognition. Additionally, SpaCy utilizes word embeddings to capture semantic information and relationships between words efficiently.

* **Testing Speed of Report Generation**

Python’s timer library was used to calculate the execution time of both systems by determine the amount of time passed between sending a request and receiving a response.

* **Testing Factual Accuracy of Generated Reports**

The information returned in the report was cross-referenced with the information in the PDFs or JSON files.

* **Testing the Lucidity of Generated Report**
* **Analyzing the Task Fit of Reports**
* **Analyzing Context Reasoning and Adaptability of the System**

**Chapter 4**

**RESULTS**

* **Semantic Similarity**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Previous Meals** | **Desired Meals** | **Age** | **Sex** | **Semantic Similarity** |
|  | **24g Jollof Rice, 68g Grilled Chicken.** | **34g Yam with Kontomire stew, 63g Soy milk** | **1-18** | **Male** |  |
|  | **44g Aprapransa, 87g Shitto,**  **83g Fried Chicken** | **126g Kenkey with Fried Fish** | **19-70** | **Male** |  |
|  | **97g Tom Brown, 136grams Almond milk, 83g Omutuo with Palmnut soup, 150g Fried Chicken, 97grams whole grain rolled oats** | **116g Akple with Okra Soup.** | **70+** | **Male** |  |
|  | **165g Akple with Okro Soup,**  **87g Grilled Chicken** | **83 g plantain with Garden egg stew, 126 Grilled Chicken** | **1-18** | **Female** |  |
|  | **52g Koko with bread, 39g Almond milk,** | **209g Plain Rice and stew, 58grams beans and fried plantain** | **19-70** | **Female** |  |
|  | **109g Aprapransa,** | **71 g Fufu with Light soup** | **70+** | **Female** |  |
|  | **63g Omotuo with groudnut soup,**  **112g Waakye with stew** | **92g Wholegrain rolled oats,**  **102g grilled chicken** | **Pregnancy** | **Female** |  |
|  | **53g Tuo Zaafi, 94g Hausa Koko with Akara and Bread.** | **53g Kokonte with groudnut soup.**  **97g Grilled Chicken** | **Lactation** | **Female** |  |

* **Mathematical Accuracy**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Previous Meals** | **Desired Meal** | **Age** | **Sex** | **KG-Augmented LLM** | **Traditional Expert System** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

* **Lucidity**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Previous Meals** | **Desired Meal** | **Age** | **Sex** | **KG-Augmented LLM** | **Traditional Expert System** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

* **Factual Accuracy**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Previous Meals** | **Desired Meal** | **Age** | **Sex** | **KG-Augmented LLM** | **Traditional Expert System** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

* **Analyzing the Task Fit of Reports**
* **Analyzing Context Reasoning and Adaptability of the System**

**General Observations**

In setting up parameters for a RAG chain, increasing the value of k, which is the number of documents utilized by the LLM, can lead to irrelevant or unclear results. It is best to minimize the value of k in setting up a RAG that uses vector data.

**Summary of Results**

The KG performs exceptionally accurately at offering specific nutritional advice about quantities of nutrients and compounds, based on a meal a user has eaten or has eaten or plan to eat since it is designed specifically for that task. However, the vector database can respond to questions both about meal content or portion analysis, and other general questions about nutrition therapy for individuals with sickle cell disease, it is not restricted to a specific question, nor is it restricted to a specific answer. In conclusion, the KG-LLM augmented system is best as a calculator for diet planning and meal watching, while the vector database is best as an all-around general sickle cell disease nutrition consultant.

**Generalization of Results**

These findings can be extended to other domains within nutrition therapy, by inferring that KG-LLMs perform better as mathematical expert systems, and VectorDatabase LLM’s perform best as general consultants.

**Limitations & Future Work**

Due to the unavailability of studies with published recommended daily intakes for sickle cell disease, BMI and activity level could not be incorporated as factors in determining nutrition therapy in the KG-Augmented LLM, However using data in the pdf, the VectorDB augmented LLM was able to reason through possible guidelines that incorporate BMI and Activity Level.

Incorporating user's BMI and activity level as well as sickle cell drug interactions is a possible area of improvement, should there be available data in the future.

**Appendix**