Retrieval-Augmented Generation Based Doctor Recommendation System Using Knowledge Graph

by

Sadia Afrose
21266004

A thesis submitted to the Department of Computer Science & Engineering in partial
fulfillment of the requirements for the degree of
M.Sc in Computer Science & Engineering

Department of Computer Science and Engineering
Brac University
August 2024

Declaration

It is hereby declared that

1. The thesis submitted is my own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.
5. I would like to request the embargo of my thesis for 24M from the submission date due to the intention of the future publication of this paper.

Student's Full Name & Signature:

Sadia Afrose
21266004

Approval

The thesis/project titled "Retrieval-Augmented Generation Based Doctor Recommendation System Using Knowledge Graph" submitted by

1. Sadia Afrose (21266004)

Of fall, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of M.Sc. in Computer Science and Engineering on November 22, 2024.

Examining Committee:

External Examiner:
(Member)

Dr. Ashis Talukder, Ph.D.
Associate Professor
Department of Management Information Systems
University of Dhaka

Internal Examiner:
(Member)

Dr. Md. Ashraful Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Supervisor:
(Member)

Md. Golam Rabiul Alam, Ph.D.
Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md Sadek Ferdous
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, Ph.D.
Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Finding suitable healthcare professionals and the growing demand for efficient healthcare access is challenging and advanced language modeling techniques to provide tailored medical advice. Reviewing the existing research on doctor recommendation systems, it became apparent that while previous authors developed functional models, their datasets are likely outdated and no longer reflective of current healthcare trends. These models would require complete retraining with a new or updated dataset to ensure their relevance and accuracy. In contrast, our approach takes advantage of the ability to update the existing database with additional, current information without the need for retraining the model from scratch. By simply integrating the updated data, we can maintain the integrity and functionality of the previous system, making our method both time-efficient and resource-conserving. This approach eliminates the need for redundant work, allowing us to leverage the previous model while ensuring the data remains current and applicable. The proposed Doctor Recommendation System leveraging a Knowledge Graph built with Neo4j and enhanced by a Retrieval-Augmented Generation (RAG) model using the LangChain framework. The system aims to provide up-to-date, personalized and accurate doctor recommendations by integrating structured and unstructured data sources. The Neo4j Knowledge Graph captures comprehensive relationships between doctors, specialties, disease, symptoms and medical conditions, offering a robust data foundation. The LangChain framework, incorporating a large language model (LLM), enhances the recommendation process by generating context-aware suggestions based on patient queries and historical data. The Doctor's Specialty Recommendation dataset has been used on three chains which are RetrievalQAChain, GraphCypherQAChain, RetrievalQAWithSourcesChain and also similarity search is used. However, based on the correctness, distance, context accuracy and CoT context accuracy, graphCypherQAChain performed well. Moreover, ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-L and BLEU are used to determine the performance for all the chains as well as compared with GPT-4. Based on these evaluation metrics, graphCypherQAChain attained a high performance level with 86% for ROUGE-1, 82% for ROUGE-2, 77% for ROUGE-3, 86% for ROUGE-L and 46% for BLEU using GPT-3.5-turbo whereas GPT-4 performed 78%, 60%, 48%, 78% and 40% respectively.

Keywords: Doctor Recommendation System, Knowledge Graph, Neo4j, Large Language Model, Retrieval-Augmented Generation, LangChain Framework, Natural Language Processing.

Table of Contents

List of Figures

List of Tables

Nomenclature

AI - Artificial Intelligence
ML - Machine Learning
KG - Knowledge Graph
LLM - Large Language Model
RAG - Retrieval-Augmented Generation
NLP - Natural Language Processing
GPT - Generative Pre-trained Transformer
LLAMA - Large Language Model Meta AI
BERT - Bidirectional Encoder Representations from Transformers
BLEU - Bilingual Evaluation Understudy
ROUGE - Recall-Oriented Understudy for Gisting Evaluation
CSV - Comma-Separated Values

Chapter 1

Introduction

Patients often face challenges in finding the right specialist for their specific medical conditions. Traditional methods of doctor referrals can be time-consuming and may not always lead to the most suitable match. An intelligent system that understands patient symptoms and dynamically recommends the best-suited doctors can significantly improve healthcare outcomes and patient satisfaction. In the proposed paper, we present an advanced method for doctor recommendation system using the power of Neo4j graph database, Large Language Models (LLM) and the LangChain framework [18] with a Retrieval-Augmented Generation (RAG) model [20], this system aims to provide personalized and efficient healthcare solutions.

The system contributes to the field of Doctor Recommendation System represents a significant advancement in the application of artificial intelligence and graph database technologies in healthcare. By integrating knowledge graph, neo4j, LLM and the LangChain framework with a RAG model, the system offers a robust solution for connecting patients with the most appropriate healthcare professionals because RAG-LLM usually performs better than LLM alone [27] [25].

The proposed approach is based on Knowledge Graphs, which represent data in a graph structure with nodes and edges, enabling the modeling of complex relationships and interactions among diverse entities. Neo4j, a leading graph database management system, provides an optimal platform for constructing and querying these knowledge graphs. By utilizing Neo4j, our system captures intricate relationships among doctors, diseases, symptoms, thereby creating a rich and interconnected data foundation. Knowledge graph (KG) based systems show promise for providing both precise and explainable recommendations due to the rich information in KG [12].

To further enhance the recommendation process with updated without retraining the entire model which will be allowing the system to integrate new knowledge dynamically [6], we integrate a Retrieval-Augmented Generation (RAG) model using the LangChain framework. RAG models combine the strengths of retrieval-based and generation-based methods, allowing for the up-to-date and dynamic generation of responses based on relevant retrieved documents. The LangChain framework facilitates the seamless integration of these models, enabling our system to generate contextually appropriate and accurate doctor recommendations.

The proposed approach combines Neo4j for data organization and LLMs within the LangChain framework to deliver relevant, personalized, and up-to-date doctor recommendations using RAG and knowledge graph. This study seeks to answer the following research questions:

RQ1. How can a knowledge graph effectively represent the relationships between doctors, diseases, and symptoms to support recommendations?

RQ2. How does the RAG model improve the accuracy and context-awareness of doctor recommendations?

RQ3. What strategies can ensure the recommendations remain up-to-date?

By addressing these questions, we aim to design a reliable and transparent system that bridges the gap between patient needs and expert medical care.

This approach offers several key advantages over traditional methods that rely solely on large language models (LLMs) trained on CSV data. The interconnected structure of the knowledge graph enables more detailed and insightful responses, going beyond the constraints of simple flat CSV data [2]. By leveraging a knowledge graph optimized for connected data, the system can provide dynamic, context-aware, and accurate recommendations based on up-to-date medical expertise and intricate symptom-disease-doctor relationships.

- The proposed system leverages the flexibility of the Neo4j knowledge graph database to enable real-time updates. This eliminates the need for frequent model retraining, ensuring that the LLM always has access to the most current information. This not only improves the accuracy and relevance of responses but also reduces the computational overhead associated with model retraining.
- The research's findings have direct implications for the medical field. By providing patients with a dedicated platform to search for doctors based on specific diseases or symptoms, the system offers a valuable tool for healthcare decision-making. This can potentially improve patient outcomes by facilitating timely access to appropriate medical care.
- The research proposes a more effective approach to representing and querying interconnected data. This performed well compared to the traditional methods relying solely on LLMs and CSV files, especially for complex queries and large datasets. The knowledge graph's ability to capture relationships and context allows for more nuanced and informative responses from the LLM.

The structure of the rest of the report is as follows. Chapter 2 reviews the related work and studies that informed and supported our research. Chapter 3 provides a detailed description of the proposed doctor recommendation system. Chapter 4 discusses the performance evaluation metrics and related analysis. Finally, Chapter 5 offers the conclusion of the paper.

Chapter 2

Related Work

There is much research being conducted around the world on recommendation systems for doctors based on the patient's symptoms. J., Liu et al. [19], this study evaluates ChatGPT's performance in recommendation tasks using a new benchmark, comparing it with traditional methods. Results show that while ChatGPT excels in rating predictions, it underperforms in sequential and direct recommendations. Despite these limitations, ChatGPT leads in generating explainable recommendations, outperforming current methods in human evaluations. This suggests its potential for improving recommendation transparency. Future research should explore better integration of user interaction data and address the gap between language models and user interests to enhance recommendation systems.

Y., Gao et al. [17] reviews Retrieval Augmented Generation (RAG), highlighting its role in enhancing large language models (LLMs) by integrating parameterized knowledge with external data. It traces RAG's evolution through three stages: Naive, Advanced, and Modular RAG, each improving on the last. The framework's integration with fine-tuning and reinforcement learning has expanded its capabilities.

P., Haque et al. [9], the author proposed a novel approach using neural network algorithms and rule inference to match patients with suitable doctors based on their symptoms. Unlike existing systems, this method doesn't require prior user interaction and achieves an impressive 96% accuracy in doctor recommendations. Future improvements include incorporating user interfaces for better usability, enabling independent user input, expanding dataset diversity, and integrating advanced techniques like fuzzy logic to enhance accuracy further.

C., Ju, & S., Zhan [11] the researcher proposed an online pre diagnosis doctor recommendation model that improves the efficiency of medical treatment seeking. By integrating ontology characteristics and disease text mining, the model enhances the accuracy of doctor recommendations based on patient descriptions and doctor specialties. Experimental validation using real internet medical data confirms the model's reliability and effectiveness. Limitations include reliance on data from a single online medical community, suggesting the need for future research across multiple platforms and cultural contexts. The study also suggests incorporating semantic and sentiment analysis techniques and user behavioral information to optimize the model further. Overall, the proposed model offers potential for broader application in online medical consultation platforms.

Chen, S. et al. [24] presents a medical question-answering system based on a knowledge graph to enhance medical services. It uses crawler technology to collect data from medical websites and builds a knowledge graph. User queries are processed through rule matching to retrieve answers, with a large model API used if rule matching fails. Built with Python, Flask, and JavaScript, the system is stable, reliable, and efficient in answering medical questions.

Wu, J. et al. [31] presents MedGraphRAG, a novel Retrieval-Augmented Generation framework that enhances Large Language Models for generating evidence-based medical responses, improving safety and reliability with private data. It overcomes traditional GraphRAG limitations by using Triple Graph Construction and U-Retrieval for better context and precision. Tested on various medical benchmarks, MedGraphRAG outperforms existing models, offering accurate, source-backed responses. The paper also compares MedGraphRAG with other RAG methods, showing its superior performance. Future work will focus on real-time data updates and clinical validation.

M., Waqar et al. [3], this study introduces a recommendation system for healthcare, utilizing an adaptive algorithm to recommend doctors based on patient preferences. Through surveys, key doctor attributes are identified and translated into a model using analytical hierarchical process (AHP). Machine learning techniques generate recommendations considering patient demographics and similarity of disease or symptoms. The system is user-friendly, incorporating maps for quick doctor location. Evaluation by experts confirms its effectiveness. Future enhancements may include incorporating patient reviews and treatment history. Integration into hospital management systems could aid in critical health situations.

S., Mondal et al. [7], the researcher proposed a novel approach to doctor recommendation systems, addressing two key challenges: modeling patient-doctor relationships efficiently and incorporating trust factors. The patient-doctor relationships are represented as a multi-layer graph, validated using a NoSQL Neo4j graph database, and compared with a relational model. Results show that the graph model outperforms the relational model due to its ability to efficiently capture complex relationships. Additionally, the paper introduces a trust factor concept, derived from database state without external information, and updated dynamically. Real-life data implementation validates the effectiveness of both the multilayer graph model and the trust factor approach.

L., Xu, et al. [32], the researcher proposed an intelligent Q&A system for Nanjing Yun- jin, leveraging Knowledge Graphs (KGs) and Retrieval Augmented Generation (RAG) technologies. It enhances information retrieval and organization, particularly with unstructured natural language data. The system utilizes text embedding models and the Euclidean distance algorithm for similarity computation, enabling accurate answers through large language models. Key achievements include semantic information extraction, efficient FAISS database construction, and enhanced generation using the LLAMA model. Despite progress, areas for improvement include data processing optimization, LLM consistency, and data security enhancements. Overall, this research advances intelligent Q&A systems for heritage preservation, with future work focusing on addressing existing challenges for broader applica-

tion. Moreover, Wang, Q. et al. [1], a helpful survey published Knowledge Graph Embedding.

Es., Shahul et al. [16] underscores the necessity for automated reference-free evaluation frameworks for Retrieval-Augmented Generation (RAG) systems. It highlights the importance of assessing three key aspects: faithfulness (whether the answer is grounded in the retrieved context), answer relevance (whether the answer addresses the question), and context relevance (whether the retrieved context is sufficiently focused). To support this, the paper introduces WikiEval, a dataset providing human judgments on these aspects. Additionally, it presents RAGAs, a framework designed to evaluate these quality dimensions effectively without requiring ground truth. Evaluations using WikiEval demonstrate that RAGAs' predictions align closely with human assessments, particularly in terms of faithfulness and answer relevance.

K. D., Spurlock et al. [30] develops an evaluation pipeline for ChatGPT as a top-n direct conversational recommendation system which emphasizes scenarios where users don't provide potential answers. Results show that re-prompting ChatGPT with feedback during interactions significantly enhances recommendation performance compared to single prompts. The study also addresses ChatGPT's popularity bias and suggests methods to counteract it, aiming to increase recommendation novelty.

Rajkumar, K. et al. [28] developed an AI-powered medical chatbots, leveraging technologies like ML, NLP, and geolocation services, are transforming healthcare by providing personalized disease diagnosis, hospital referrals, and user support. These chatbots operate in both online and offline modes, with the offline mode offering automated responses based on trained data. Continuous training and testing are essential for integrating new data, ensuring the chatbot remains accurate and up-to-date with the latest medical knowledge. Future work focuses on enhancing the system with telemedicine integration, expanding its reach to global users, and improving the accuracy of diagnoses through advanced ML models. Ethical concerns, such as data privacy and regulatory compliance, also need to be addressed to ensure secure and responsible use. This evolving technology promises to revolutionize healthcare by making services more accessible, efficient, and personalized.

L., Chen et al. [15] found that GPT-3.5 and GPT-4 exhibited significant changes in behavior over a short period. This highlights the need for ongoing monitoring of LLM drift in real-world applications. The study also revealed the difficulty of uniformly improving LLMs. Fine-tuning for specific tasks can sometimes lead to decreased performance on other tasks. Moreover Zhao, Z. et al. [33], also showed that how Recommender Systems (RecSys) are crucial for providing personalized suggestions in e-commerce and web applications. While Deep Neural Networks (DNNs) have advanced these systems, they struggle with capturing textual information and generalizing across scenarios. The rise of LLMs like ChatGPT and GPT-4, with their strong language understanding and generalization abilities, has led to efforts to integrate them into RecSys.

Also, K., Pichai [21] emphasized the effectiveness of a modular RAG approach over traditional fine-tuning, showcasing its potential for optimizing Large Language Mod-

els (LLMs) with tailored datasets and also stated the importance of context-aware fine-tuning in achieving high linguistic precision. However, according to Siriwardhana, S. et al. [22], fine-tuning tends to be more costly due to the extensive data it requires, its substantial computational demands, and the longer time needed for training.

2.1 Background Study

This section offers valuable supplementary material to help understand the functioning of the doctor recommendation system. It also includes various equations and figures to enhance comprehension of the concepts presented.

2.1.1 Knowledge Graph

A knowledge graph is a data structure that represents relationships between entities in a graph format, where entities (such as people, places, or concepts) are nodes, and the relationships between them are edges. It organizes information in a way that enables machines to understand and infer complex relationships, providing a semantic layer to raw data. Knowledge graphs are used in various applications, such as search engines, recommendation systems, and AI-based decision-making, by capturing and integrating data from diverse sources, making it easier to discover insights and make inferences based on context.

Components

The primary components are (1) Nodes: Represent entities or concepts for instance people, places, things. (2) Edges: Represent relationships or connections between entities for instance works, treats. (3) Properties: Attributes or additional information associated with nodes or edges for instance a person's age, the weight of an object, or the date of a relationship. These elements together form the graph structure that captures and represents knowledge in a machine-readable way.

Types of Knowledge Graph

Various knowledge graph models exist, including Property Graph Model, Directed Graphs, Undirected Graphs, Weighted Graphs, Acyclic Graphs, Cyclic Graphs, Hypergraphs, and Multigraphs. For the research, the Property Graph Model is adopted. This model is particularly suitable for this application due to its flexibility in representing complex relationships and its ability to handle diverse data types.

Ontology

An ontology is a formal, structured framework that defines the concepts, entities, relationships, and rules within a specific domain of knowledge. It provides a shared vocabulary and a clear description of how entities are related to one another, enabling consistent data representation and reasoning. Ontologies are used to model knowledge in a way that both humans and machines can understand, often serving as the foundation for knowledge graphs technologies.

2.1.2 Neo4j

Neo4j is a popular open-source and powerful graph database management system specifically designed for storing and querying knowledge graphs [10]. It provides a flexible, scalable platform for building and working with knowledge graphs. Neo4j uses the Cypher query language, which allows for expressive and intuitive querying of graph data. Known for its high performance and ability to handle complex relationships, Neo4j is widely used in various fields, including social networks, fraud detection, and recommendation engines, enabling organizations to gain deeper insights by analyzing interconnected data. The way Neo4j interacts with and manages graph data can be visualized by the equation:

$$CypherQuery(Pattern, Return) = ResultSet \tag{2.1}$$

This equation illustrates how Cypher which is a the query language used in Neo4j processes patterns and returns corresponding results.

Graph Query

Graph queries refer to the process of querying or retrieving data from a graph database or graph data structure, where the data is represented in the form of nodes (entities) and edges (relationships). In graph databases, queries are used to explore relationships between entities, find patterns, and retrieve specific information based on how the data is connected. It consists of three elements. (1) Nodes which represent entities. (2) Edges which represent relationships between nodes. (3) Properties which are attributes or values associated with nodes or edges.

2.1.3 Large Language Model (LLM)

Large Language Model also known as LLM. It is a type of artificial intelligence (AI) designed to understand and generate human-like text based on vast amounts of data to further refine user queries and enhance the LLM's understanding

An LLM is great for generating general-purpose text, creative content, and answering questions based on its training, but it's limited by the information it was trained on. Simple LLM workflow is shown in Figure 2.1 based on the above context.

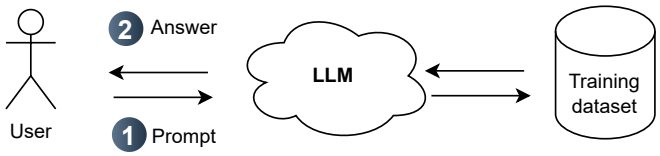

Figure 2.1: Architecture of Large Language Model (LLM) [14]

Examples like GPT which is a generative Pre-trained Transformer, BERT, and T5 showcase their versatility and effectiveness across diverse tasks.

LLM generates text based on patterns and information it has learned during training. However, it shows a fixed knowledge cutoff (e.g., GPT-3's knowledge is

only up to 2021) and can't access any new or real-time information that was not part of their training data.

The transformer architecture, which is the basis for many LLMs like GPT, uses the self-attention mechanism. The main equation for computing the attention scores is:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.2}$$

Where:

- Q = Query matrix,
- K = Key matrix,
- V = Value matrix,
- d_k = Dimensionality of the key vectors.

While LLMs can generate text with high fluency and coherence, they can hallucinate or make up information, especially if they are asked about very specific, niche, or recent topics. They cannot directly query external sources or databases to get up-to-date information.

2.1.4 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation is a sophisticated technique that boosts the performance of language models by embedding external information retrieval into the text generation process. This method synergizes the strengths of both information retrieval systems and generative language models, thereby producing responses that are more precise and contextually appropriate. The RAG framework emerges as a key tool for enhancing LLMs, promising significant improvements in handling diverse linguistic and cultural contexts [21]. The RAG methodology includes two main components - retrieval and generation.

Retrieval

This part is responsible for retrieving relevant documents or pieces of information from a large corpus or database. It typically uses search techniques like vector similarity or semantic search to find the most relevant information based on the user's query. This retrieval process is typically managed by a separate model, such as a dense passage retriever [5] or a traditional search engine. Dense passage retrievers are specifically designed to efficiently retrieve relevant passages from large text corpora based on semantic similarity. These models use techniques like vector embeddings and similarity measures to identify passages that are most relevant to a given query.

Generation

This is usually a Large Language Model (LLM), such as GPT, T5, or BART, that takes the retrieved information and generates an appropriate, coherent and informative response based on that data. Once relevant information has been retrieved, this generation component gets in action.

The RAG model can be configured to incorporate different retrieval strategies and generative models, allowing for customization to specific tasks and requirements. By combining the strengths of information retrieval and text generation, RAG models can produce more accurate, informative, and contextually relevant responses compared to traditional language models.

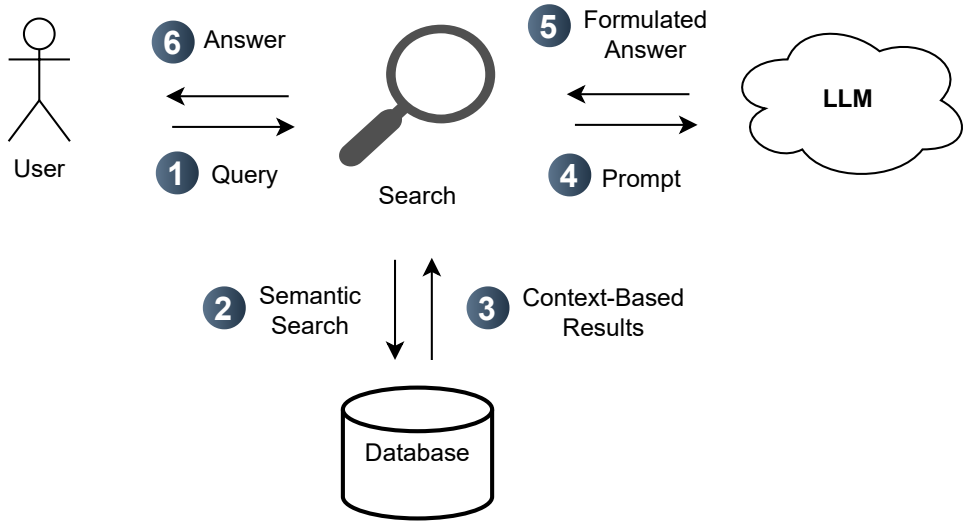

Figure 2.2: Architecture of Retrieval-Augmented Generation (RAG) [14]

In Figure 2.2 shows a typical RAG workflow, a user first inputs a query or prompt. The system then retrieves relevant documents from a designated corpus, using techniques like semantic search to ensure accuracy. These retrieved documents are used as context for the generative model, providing it with valuable information to draw upon. Finally, the generative model produces a well-informed response based on the query and the retrieved context, ensuring that the output is both relevant and informative. General equation used by RAG can be illustrated as:

$$\text{RAG(prompt)} = \text{f (R(p),Understand(R(p)), Generate(R(p),Understand(R(p))))} \tag{2.3}$$

where

- R(p) = Retrieve(prompt)

Retrieval-Augmented Generation (RAG) offers a significant advancement in language model capabilities. By accessing and incorporating relevant information from external sources, RAG models can significantly expand their knowledge base, reducing the risk of generating misinformation. This enhanced access to up-to-date data allows RAG models to provide more accurate, informative, and contextually relevant responses, making them invaluable tools for various applications, particularly in academic research and professional settings. This approach is especially beneficial in applications like recommendation systems, question answering, social network analysis [8] where the model can access a vast corpus of knowledge to provide accurate and informative responses.

2.1.5 LangChain

LangChain is a popular framework that simplifies the implementation of RAG. It allows to connect retrieval and generation components, streamlining the process of building RAG systems.

A simplified LangChain implementation involves a two-step process. First, a retrieval model is used to identify relevant documents based on the given query. These documents are then fed into a language model, which generates a response by incorporating the retrieved information as context. This sequential approach, known as a pipeline, effectively integrates information retrieval and generation capabilities within a single framework, making it a versatile tool for various natural language processing tasks.

Retrieval Augmented Generation (RAG) played a crucial role in augmenting the LLM's understanding. By retrieving relevant information from the Knowledge Graph and other data sources, RAG provided the LLM with additional context, enabling it to generate more accurate and informative responses. This integration significantly improved the system's ability to address complex queries and provide comprehensive answers.

There are various types of implementations of chains. In this research, three of these have been used.

RetrievalQAChain

A chain that retrieves relevant documents or information and then uses a language model to answer questions based on that retrieved information. It first retrieves relevant documents or data based on the query and then passes this information to the LLM, which generates a response based on the retrieved content.

GraphCypherQAChain

A chain that queries a graph database (like Neo4j) using Cypher queries and then uses the results to answer questions [29]. The LLM formulates a Cypher query based on a user's question, retrieves the relevant data from the graph, and then processes that data to provide a natural language response.

RetrievalQAWithSourcesChain

A variation of RetrievalQAChain that not only provides an answer but also includes the sources of the information used to generate the answer. So it is an extension of the RetrievalQA chain, where the answer is not only returned but also accompanied by the source(s) of the information used to generate the answer. This chain makes it possible to provide citations or references, so users know where the information came from. This uses contextualized prompts [13] to provide more accurate and contextually relevant responses.

Similarity Search is not itself a chain but a method used to find and retrieve documents or passages that are similar to a given query based on vector embeddings or other similarity measures. This is commonly used in the retrieval components of chains. Vector embeddings are numerical representations of text data generated by the OpenAI model, are stored in a Neo4j database to facilitate efficient data retrieval and analysis.

2.1.6 GPT-3.5-turbo

GPT-3.5-turbo is a highly optimized variant of OpenAI's GPT-3, designed as part of the broader class of Large Language Models (LLMs). It maintains the core capabilities of GPT-3, such as text generation, question answering, and summarization, but is more efficient in terms of both cost and performance. With faster response times and lower deployment costs, GPT-3.5-turbo is ideal for applications like chatbots, content generation, and real-time text processing. It leverages the transformer architecture common to all LLMs but with optimizations that make it particularly resource-efficient. In Retrieval-Augmented Generation (RAG) systems, GPT-3.5-turbo can be used as the generator, combining retrieved information with its generative capabilities to produce relevant and coherent responses. Its balance of efficiency and performance makes it a key tool in large-scale, real-time LLM applications where low-latency and high-quality output are essential.

2.1.7 Python

The primary programming language for implementing and integrating all components. Visual Studio Code is used for programming all of our research work.

Chapter 3

Proposed Doctor Recommendation System

In the realm of healthcare, patient-centered care has become increasingly essential. A critical component of this approach is the ability to efficiently and accurately connect patients with the most suitable medical specialists. Traditional methods of doctor referral often prove inadequate, as they can be time-consuming and may not always yield the best match. To address these challenges, the proposed research paper introduces a Doctor Recommendation System that leverages the power of a Knowledge Graph and RAG model for providing personalized, efficient, and accurate doctor recommendations revolutionizing the way patients access healthcare services.

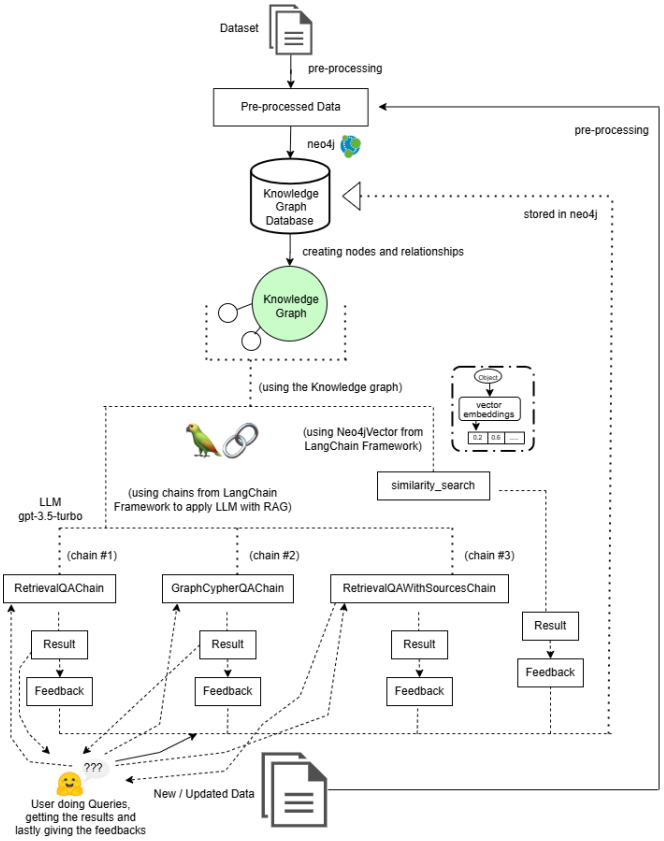

Figure 3.1: Top level overview of the proposed system

3.1 Dataset Description

The dataset [34] consists of five CSV files (Disease_Descrip-tion, Doctor_Specialist, Doctor_Versus_Disease, Original_Dataset, Specialist, Symptom_Weights), providing a rich source of information on symptoms, diseases, specialist doctors, and associated weights. This dataset is well suited for applying various data cleaning techniques and enables the development of predictive models. To ensure data consistency and facilitate analysis, the dataset was assigned column names, as the original data lacked these essential identifiers. Additionally, the second dataset [35] was incorporated to enhance the system's updating capabilities. By combining these datasets, we established a comprehensive foundation for our research, enabling us to explore the relationships between diseases, symptoms, and specialist doctors and develop effective recommendation models.

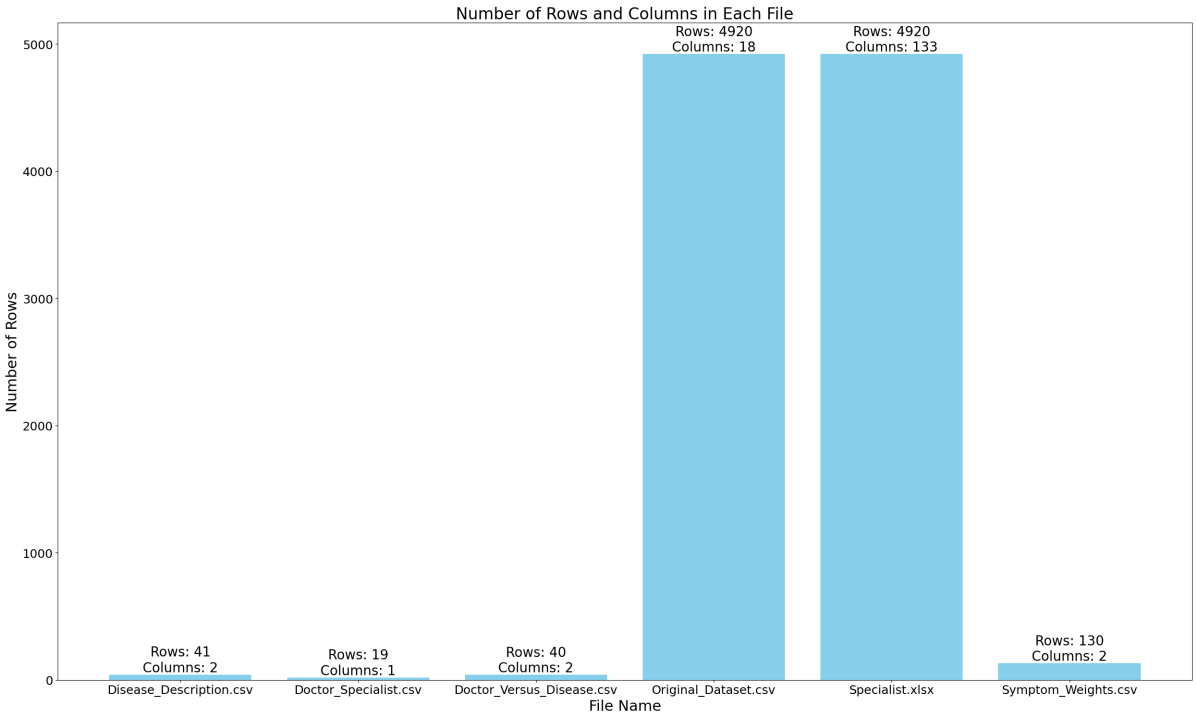

Figure 3.2: Doctor's Specialty Recommendation dataset frequency

As shown in the graph, Disease_Description has 41 rows and 2 columns, Doctor_Specialist has 19 rows and 1 column, Doctor_Versus_Disease has 40 rows and 2 columns, Original dataset has 4920 rows and 18 columns, Specialist has 4920 rows and 133 columns, Symptom_eights has 130 rows and 2 columns. Hence, we have enough data to perform our research.

3.2 Preprocessing

To streamline the data analysis process, all datasets were initially imported into Python DataFrames, utilizing the Pandas library for its robust data handling capabilities. For datasets that did not come with predefined headers, appropriate column names were assigned to ensure uniformity and data integrity. For example,

the DataFrames for symptom weights and doctor versus disease mappings were standardized with headers such as ["Symptom", "Weight"] and ["Disease", "Doctor"], respectively. This standardization was pivotal in maintaining a structured dataset, which facilitated accurate data manipulation and ensured consistency across the analytical process.

Furthermore, to enhance data retrieval efficiency, the Disease_Description and Symptom_Weights DataFrames were converted into dictionaries. This transformation utilized the dictionary data structure's advantage of average-case constant time complexity for key-based access, which significantly accelerated data retrieval operations. By converting these DataFrames into dictionaries, we enabled rapid and efficient access to disease descriptions and symptom weights based on their corresponding keys. This approach not only optimized the performance of data retrieval but also supported effective management of large datasets and complex analytical tasks. The use of dictionaries thus played a critical role in improving the overall efficiency and speed of the data analysis workflow. .

3.3 Knowledge Graph Construction

This allows us to represent complex data with structured entities and relationships, with ontologies adding a formal schema to define the meaning of each node and connection. This structure simplifies the integration of diverse data sources and enables efficient querying of relationship-rich data in Neo4j

3.3.1 Knowledge Graph

The concept of Knowledge Graphs is a term popularized by Google [4]. Our knowledge graph comprised of three node types: diseases, doctors, and symptoms. These nodes were interconnected through relationships representing associations between diseases and symptoms, diseases and doctors, and symptoms and diseases. To represent the strength of these associations, we incorporated weights as properties of relationships. Property Graph Model is used which is particularly suitable for our application due to its flexibility in representing complex relationships and its ability to handle diverse data types.

3.3.2 Ontology

Define the ontology with concepts such as "Disease", "Doctor", "Symptom", and appropriate relationships like "HAS_SYMPTOM", "TREATS". The weights and other relevant attributes are incorporated into the ontology such as "name", "weight", "description", "embedding" etc. which are also known as property keys. Ontology is essential for ensuring that the data within the knowledge graph is interpreted consistently [26].

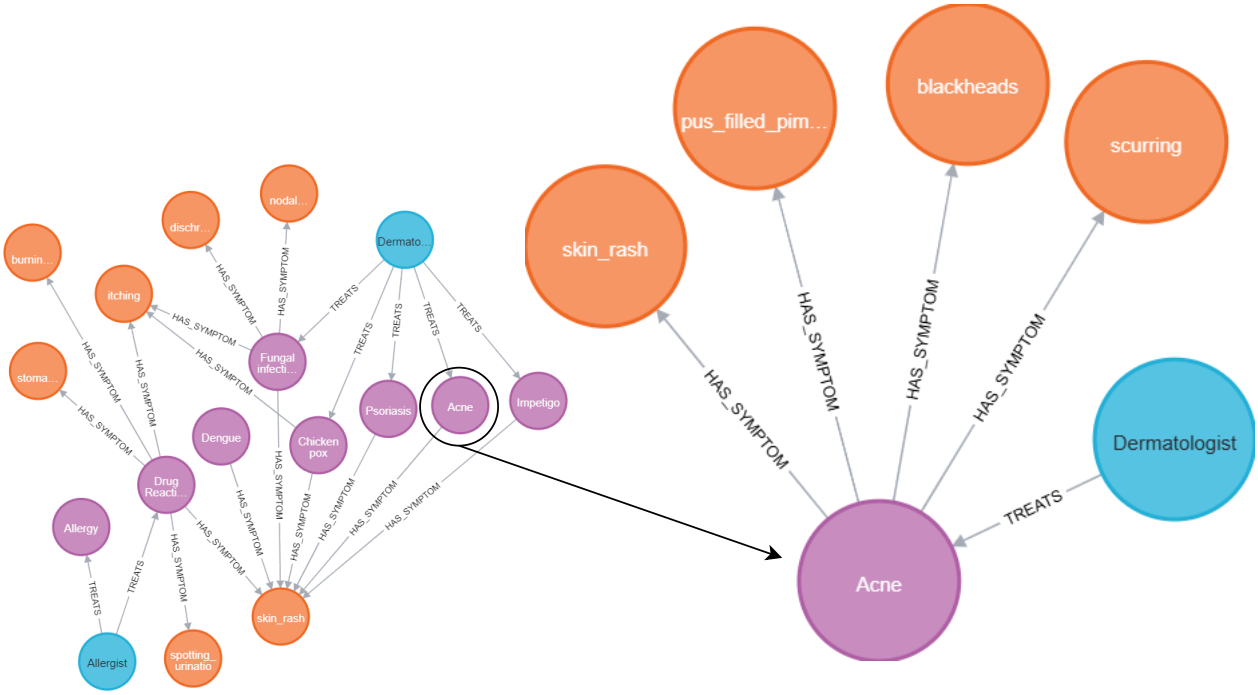

Figure 3.3: A fusion of both "TREATS" and "HAS_SYMPTOM" relationship among specialists, symptoms and diseases

In Figure 3.3, There are two types of relationships. "HAS_SYMPTOM" relationship between diseases and symptoms nodes and "TREATS" relationship between specialist and disease nodes. All of the nodes are connected undirected-wise with these two relationships,

3.4 LLM Integration with RAG

This involves combining the generative power of LLMs with external knowledge sources to improve the quality of answers. LangChain provides a flexible way to manage the retrieval and generation process by seamlessly integrating retrieval systems like knowledge graphs with language models.

3.4.1 Embedddings

In the context of a RAG model, embedding plays a key role in enabling the model to efficiently retrieve relevant information from a large corpus of text. The most common way to measure similarity between embeddings is the cosine similarity, which is calculated as:

$$\text{Cosine Similarity} = \frac{A \cdot B}{\|A\|\|B\|} \tag{3.1}$$

where A and B are the embedding vectors for two texts. Cosine similarity yields a value between -1 and 1, where 1 indicates perfect similarity, 0 indicates no similarity,

and -1 indicates perfect dissimilarity. For retrieval, only positive similarity scores are relevant, as they indicate semantic closeness.

OpenAIEmbeddings

OpenAIEmbeddings refers to a specific implementation of embeddings based on OpenAI's models, such as GPT-3, GPT-4, or other language models that can generate vector representations of text. These embeddings are used to convert text into numerical vectors that capture the semantic meaning of the text. It is often used to access and work with models like text-embedding-ada-002 which is developed by OpenAI for generating high-quality, dense vector embeddings of text.

OpenAIEmbeddings, like other embeddings, aim to minimize the loss function during training. This can include cross-entropy loss for classification tasks, or mean squared error for embedding alignment tasks. Let's say for an embedding v_i for input i, and v_j for input j, we want them to have high similarity if they are semantically related. If v_i and v_j are expected to be close, the model will train to minimize the *cosine distance* between v_i and v_j., defined as:

$$\text{Cosine Distance} = 1 - \text{Cosine Similarity} \tag{3.2}$$

3.4.2 ChatOpenAI

ChatOpenAI refers to OpenAI's conversational models, like GPT-3 and GPT-4, designed to generate human-like, contextually aware responses in a dialogue format. Unlike OpenAIEmbeddings, which convert text into vectors to measure semantic similarity, ChatOpenAI generates natural language responses based on the entire conversation context. It uses a sequence of user inputs, assistant replies, and system instructions to maintain coherence over multiple turns.

The model is trained using cross-entropy loss, where it minimizes the difference between predicted and actual tokens to improve response quality. While embeddings focus on calculating semantic similarity (often with cosine similarity), ChatOpenAI focuses on generating appropriate, fluent replies, making it ideal for applications like chatbots and virtual assistants.

$$\text{Cross-Entropy Loss} = -\sum_i p_{\text{actual},i} \cdot \log(p_{\text{predicted},i}) \tag{3.3}$$

where $p_{\text{actual},i}$ is the actual probability distribution of the token at position i, and $p_{\text{predicted},i}$ is the predicted probability distribution from the model. This loss function helps the model improve its language generation capabilities, ensuring that the model produces more relevant and coherent responses in a conversation. In contrast to embedding models, which focus on measuring similarity between vectors, ChatOpenAI focuses on generating contextually relevant and fluent text based on conversation history.

3.4.3 Prompt

Prompt refers to the text or query that is provided to an LLM in order to generate a response or perform a specific task. Prompts are essential for controlling the

behavior of an LLM, as the way a prompt is structured can significantly influence the quality and relevance of the output. LangChain offers a flexible and modular approach to prompt engineering, allowing developers to customize and optimize prompts for a variety of applications.

Three types of prompts have been used.

ChatPromptTemplate

ChatPromptTemplate is used to create prompts for chat-based interactions, typically involving multiple turns between a user and the system. This template supports the management of both user and system messages, helping to construct more natural, multi-turn conversations. It is ideal for scenarios where it is needed to simulate a back-and-forth exchange like a chatbot.

HumanMessagePromptTemplate

HumanMessagePromptTemplate is specifically designed for the user's input in a conversation, this template formats prompts that represent messages coming from a human user. It helps distinguish the user's messages from the system's responses in a structured way. This is useful when it is interacting with a model in a conversational context where the human's message needs to be formatted clearly.

SystemMessagePromptTemplate

SystemMessagePromptTemplate is used for creating system messages in a chat based interaction. System messages define the model's behavior or give it specific instructions about how to respond, often setting the tone, context, or style of the conversation. This template is used to guide the system in how to interpret user queries or to set the overall context for the conversation.

3.4.4 User Queries

To enhance user interactions, we integrated Large Language Models into our system. LLMs, known for their ability to understand and generate human language, played a key role in interpreting user queries. By processing inputs, they captured intent and context, enabling accurate and relevant responses. This capability allowed users to ask natural language questions, such as "What doctor specializes in X disease?" or "What are the symptoms of Y disease?", greatly improving the user experience.

To further improve the accuracy and relevance of responses, we implemented Retrieval-Augmented Generation (RAG) using LangChain. When a user query is received, the system first retrieves relevant information using the RetrieverQA chain, which queries a database or knowledge source. This is followed by CypherQAChain, which applies advanced reasoning over graph data, and RetrievalQAWithSourcesChain, ensuring that the generated response is both relevant and grounded in verifiable sources. Finally, a generation step uses an LLM to craft a coherent and contextually appropriate answer, incorporating both the retrieved knowledge and the user's query.

We used GPT-3.5-turbo, an open-source LLM, to ensure flexibility and avoid vendor lock-in, while maintaining control over the system's architecture.

3.5 Dynamic Updates and Continuous Learning

This refers to systems that can adapt and improve over time by incorporating new data without requiring complete retraining from scratch.

3.5.1 Data Updates

Regularly update the knowledge graph and ontology with new data, ensuring that the system stays up-to-date with the latest medical information.

3.5.2 Feedback Loop

This feature is not implemented in this research and kept it as a future implementation. We can implement a feedback mechanism where user interactions and feedback are collected using reinforcement learning with feedback from a RAG model and incorporating straightforward techniques to improve robustness against noisy web content and conversational variability [23]. Using this data to refine the recommendation algorithms, improve the knowledge graph, and enhance LLM understanding.

3.5.3 Fine-Tuning

This feature is also not implemented in this research and kept it as a future implementation due to cost constraints.

Chapter 4

Result and Discussion

The proposed technique has been tested on various data sets based on doctor, disease and their symptoms. The recommendations, implemented with three types of chains which are RetrievalQAChain, graphCypherQAChain, RetrievalQAWithSourcesChain from the Langchian framework in order to impose the RAG model on our generated knowledge graph. All the computations have been performed on Intel Core i3 with 3.4 GHz processor and 8 GB RAM.

4.1 Performance Evaluation Metrics

LangSmith is a development and debugging tool within the LangChain ecosystem that helps developers build, test, and refine applications leveraging Large Language Models (LLMs) and chain-based workflows. BLEU is a precision-focused metric for evaluating machine translation by comparing predicted text with reference text using n-gram overlap, while ROUGE is a recall-focused metric for evaluating text summarization by assessing n-gram recall, precision, and overlap between predicted and reference summaries.

4.1.1 Correctness

In the context of LangSmith, correctness refers to the ability of a language model to produce outputs that are accurate, reliable, and meet the expectations of the task or prompt. LangSmith is a framework designed to help to build, test, and fine-tune language models, and correctness plays a central role in evaluating the performance of the models.

The correctness of the model's responses is quantified using the following equation:

$$Correctness = (NoQAC/TNoQ) * 100 \tag{4.1}$$

where,

- NOQAC = Number of Question Answered Correctly
- TNOQ = Total Number of Questions

In this equation "Number of questions answered correctly" refers to the count of questions for which the model's generated answers exactly match the reference answers provided by human evaluators. This metric is essential for assessing the accuracy of the model's responses. And "Total number of questions" denotes the total count of questions that were evaluated. This value serves as the denominator in the formula and provides context for the proportion of correctly answered questions.

4.1.2 Contextual Accuracy

In the context of LangSmith, Contextual Accuracy refers to the model's ability to maintain coherence and relevance in its responses while considering the full context of a task or a conversation. LangSmith is a framework designed to evaluate and test language models, and contextual accuracy is one of the key aspects that determines how well a model performs in real-world use cases, especially for conversational AI or multi-step reasoning tasks. The Contextual Accuracy of a natural language processing (NLP) model can be defined as a function of three key components: Response Relevance, Context Understanding, and Factual Accuracy. This relationship is expressed as:

$$ContextualAccuracy = f(RR, CU, FA) \tag{4.2}$$

where,

- RR = Response Relevance
- CU = Context Understanding
- FA = Factual Accuracy

"Response relevance" evaluates how closely a generated reply aligns with the input query, encompassing semantic similarity (meaning alignment) and topic coherence (staying on-topic). Human judgment plays a key role in assessing the overall relevance. "Context understanding" which is effective involves the model recognizing key entities and maintaining the correct topic throughout a conversation, ensuring responses are relevant and informative. "Factual accuracy" is vital in research, achieved through comparing responses against reliable external sources and consulting domain experts to ensure the information is credible and precise.

4.1.3 CoT Contextual Accuracy

In the context of LangSmith, CoT Contextual Accuracy (Chain-of-Thought Contextual Accuracy) measures how well a language model maintains logical coherence and relevance across multiple reasoning steps, especially for tasks that require complex or multi-step problem-solving. Chain-of-Thought (CoT) reasoning involves breaking down a problem into intermediate steps rather than providing a direct answer, allowing the model to handle tasks like arithmetic, decision-making, or multi-step questions more effectively. CoT Contextual Accuracy evaluates whether the model's reasoning is logically consistent, contextually appropriate, and aligned with the task requirements. It ensures that the model's chain of thought not only leads to the

correct final answer but also remains coherent and relevant throughout the process, adhering to the context provided in the task.

the used equation is as follows:

$$CoTContextualAccuracy = f(RR, CoTRQ, FA) \tag{4.3}$$

where,

- RR = Response Relevance
- CoTRQ = CoT Reasoning Quality
- FA = Factual Accuracy

"Response Relevance" measures how well a generated response aligns with the input context, assessed through metrics such as semantic similarity, topic coherence, and human judgment. "CoT Reasoning Quality" evaluates the model's ability to break down complex problems into logical, sequential steps, ensuring clarity and relevance in the intermediate reasoning, as well as the overall coherence of the thought process. "Factual Accuracy" measures the correctness of the information in the response, often by comparing it to reliable external sources or consulting domain experts.

4.1.4 Jaro-Winkler Distance

In th context of LangSmith, the Jaro-Winkler distance is often used to evaluate the similarity between two strings, particularly for tasks like measuring text similarity or fuzzy matching in natural language processing (NLP). It is a string metric that calculates the similarity between two strings based on the number and order of matching characters. This metric is useful in assessing how similar two strings are, which can be applied to evaluating response relevance, semantic similarity, or context understanding.

The Jaro-Winkler distance comparison is a method for measuring the similarity between two strings. It is a variation of the Winkler distance, which is itself a modification of the Levenshtein distance. The general equation for the Jaro-Winkler distance is:

$$d(s1, s2) = f(LD(s1, s2) - (lmin(length(s1), length(s2))similarity) \tag{4.4}$$

where,

- d(s1, s2) is the Jaro-Winkler distance between strings s1 and s2.
- LD(s1, s2) is the Levenshtein distance between strings s1 and s2
- l is a scaling factor (typically set to 0.1).
- min(length(s1), length(s2)) is the minimum length of the two strings.

- similarity is a measure of the similarity of the initial characters of the two strings. It is typically

Similarity is calculated as follows:

$$\text{similarity} = \begin{cases} 0, \text{if the first } j \text{ characters of } s1 \\ \quad \text{and } s2 \text{ are not equal,} \\ \\ 1 - (j/max(length(s1), \\ \quad length(s2))), \text{otherwise} \end{cases} \tag{4.5}$$

Where:

- j is the number of characters in the longest common prefix of $s1$ and $s2$.

The Jaro-Winkler distance is a more sensitive measure of similarity than the Levenshtein distance, as it takes into account the similarity of the initial characters of the two strings. This can be useful for applications where it is important to distinguish between strings that are similar in their initial characters but different in their overall content.

4.1.5 BLEU

Bilingual Evaluation Understudy (BLEU) is a metric used to evaluate the quality of machine-generated translations compared to human-generated reference translations. It measures the precision of n-grams (sequences of n words) in the candidate translation relative to the reference translations, with a penalty for translations that are too short.

The BLEU score is calculated as a weighted average of precision scores for different n-grams (typically up to 4-grams). The formula consists of two components: precision and brevity penalty. Equation for BLEU:

$$\text{BLEU}(p_1, p_2, \ldots, p_n) = BP \times \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{4.6}$$

Where:

- p_n is the precision for n-grams (precision for unigrams, bigrams, trigrams, etc.).
- w_n is the weight for n-grams (typically $w_n = \frac{1}{N}$).
- N is the maximum n-gram order (usually 4).

Brevity Penalty (BP) is applied to prevent very short translations from receiving a high score. It is calculated as:

$$BP = \begin{cases} 1 & \text{if } c > r \\ \exp\left(1 - \frac{r}{c}\right) & \text{if } c \leq r \end{cases} \tag{4.7}$$

Where:

- c is the length of the candidate translation (number of words).
- r is the length of the reference translation (number of words).

4.1.6 ROUGE

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) is a set of metrics used for the automatic evaluation of machine-generated text, particularly for tasks like text summarization and machine translation. It compares the n-grams (or other components like word sequences) of the machine-generated output against one or more reference outputs (human-written) to measure the quality of the generated text.

Unlike BLEU, which focuses on precision, ROUGE focuses on recall. ROUGE can be used to evaluate a variety of tasks like summarization, sentence extraction, and translation, and it is widely adopted in Natural Language Processing (NLP). ROUGE calculates recall, precision, and F1-score so general ROUGE formulas (for n-grams):

$$\text{Recall} = \frac{OCR}{NR} \tag{4.8}$$

Where:

- OCR is Number of overlapping n-grams in candidate and reference.
- NR is Number of n-grams in reference.

$$\text{Precision} = \frac{OCR}{NC} \tag{4.9}$$

Where:

- OCR is Number of overlapping n-grams in candidate and reference.
- NC is Number of n-grams in candidate.

$$\text{F1 Score} = \frac{2 \times P \times R}{P + R} \tag{4.10}$$

Where:

- P = Precision
- R = Recall

ROUGE-1

ROUGE-1 refers to the overlap of unigrams means single words. The ROUGE-1 score is calculated as follows:

$$ROUGE - 1 = \frac{count(GiS \cap GiR)}{count(GiR)} \tag{4.11}$$

where,

- GiR = Grams in Summary
- GiR = Grams in Reference

count(GiS $\cap$ GiR) or count(grams in summary $\cap$ grams in reference) is the number of unigrams that appear in both the generated summary and the reference summary.

ROUGE-2

ROUGE-2 refers to the overlap of bigrams means pairs of words. The formula used for calculating ROUGE-2 score is:

$$\text{ROUGE-2} = \frac{(countofBCtobothSaR)}{(countofbigramsintheRS)} \tag{4.12}$$

where,

- BC = Bigrams Common
- SaR = Summary and Reference
- RS = Reference Summary

ROUGE-3

ROUGE-3 measure the overlap of trigrams means sequences of three consecutive words. The Rouge-3 score is calculated as follows:

$$\text{Rouge-3} = maxF1 - score(n - gram) \tag{4.13}$$

where,

- n-gram represents a sequence of n words (in this case, n=3).
- F1-score(n-gram) is the harmonic mean of precision and recall for the n-grams.

ROUGE-L

ROUGE-L measures the longest common subsequence (LCS) between the reference and candidate texts. It captures the longest sequence of words that appear in both the reference and the candidate, preserving their order. The equation based on which ROUGE-L is calculated is:

$$\text{ROUGE-L} = \frac{LCS(S, R)}{len(R)} \tag{4.14}$$

where,

- LCS(S, R) is the length of the longest common subsequence between the system-generated summary S and the reference summary R.
- len(R) is the length of the reference summary.

4.2 Result Analysis

RetrievalQAChain, graphCypherQAChain, RetrievalQAWithSourcesChain are evaluated in LangSmith using metrics like Correctness, Context accuracy, CoT Correctness accuracy, and Jaro-Winkler distance, followed by BLEU and ROUGE for additional evaluation. Finally, we compare our work with GPT-4.

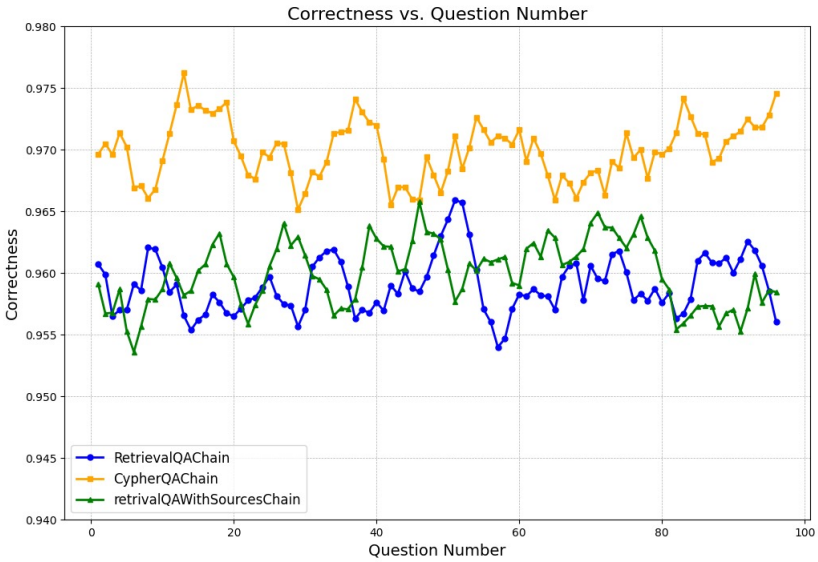

Figure 4.1: Model performance comparison: Correctness vs. Question Number

In Figure 4.1, the result shows that among all chains graphCypherQAChain performed incredible with almost 98%.

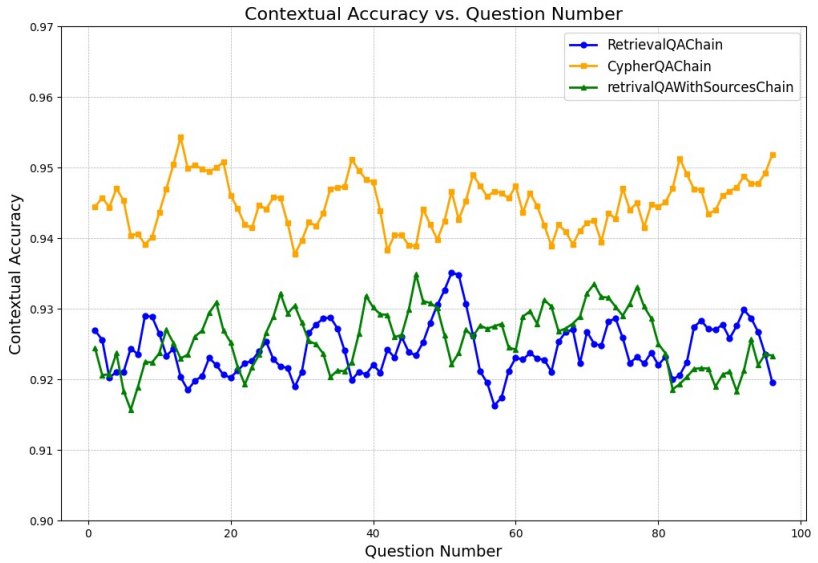

Figure 4.2: Contextual Accuracy evolution over Questions

In Figure 4.2, the result shows that among all chains graphCypherQAChain performed incredible with almost 95%.

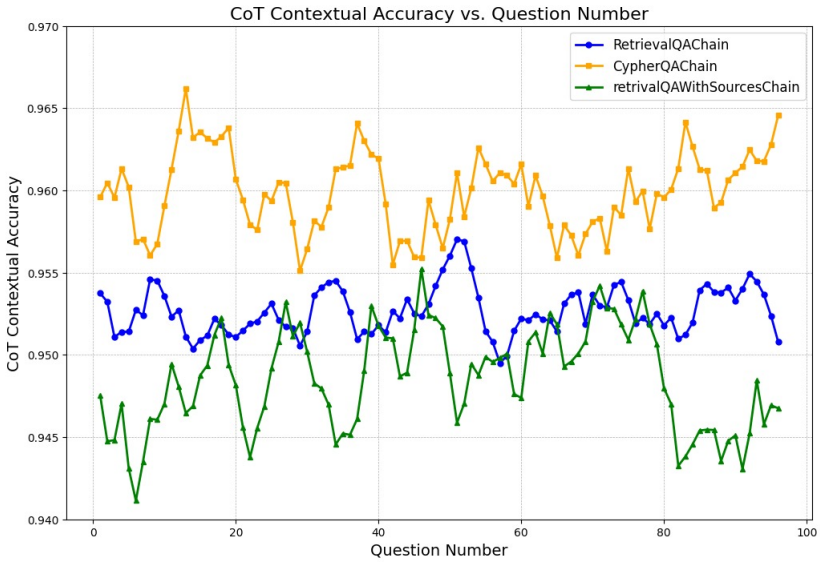

Figure 4.3: CoT Contextual Accuracy trends: all chains comparison

In Figure 4.3, the result shows that among all chains graphCypherQAChain performed incredible with almost 97%.

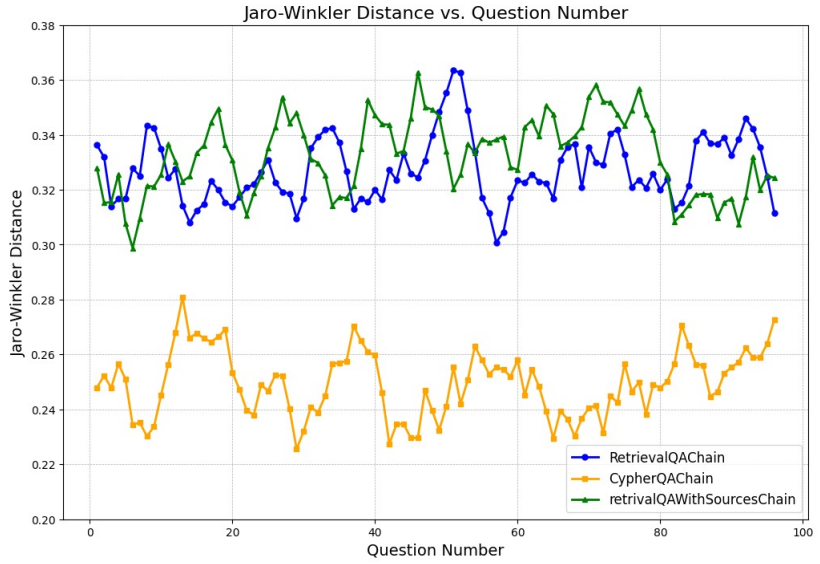

Figure 4.4: Jaro-Winkler Distance comparison: all chains performance

In Figure 4.4, the result shows that among all chains graphCypherQAChain performed incredible with almost 22% distance.

100 questions were used to find all the accuracies below.

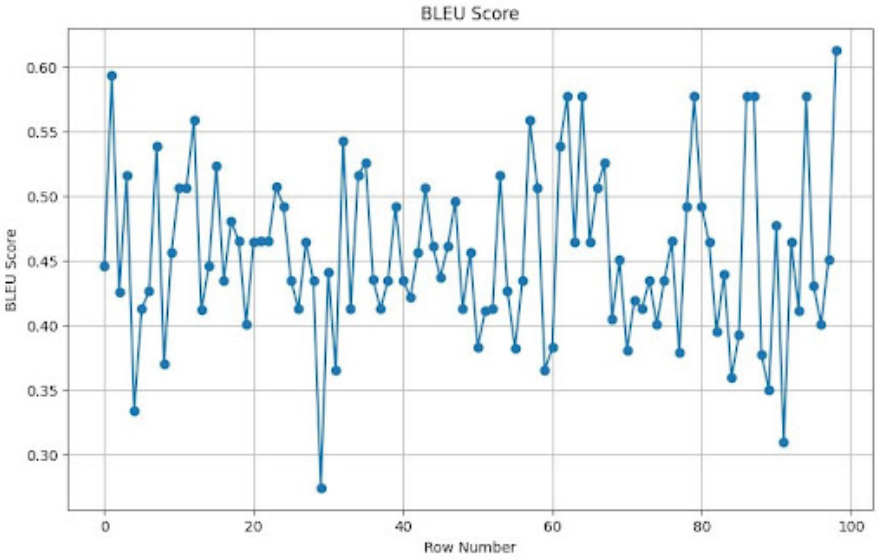

Figure 4.5: BLEU score of GraphCypherQAChain recommendations

In Figure 4.5, the result shows that the accuracy we got for BLEU on GraphCypherQAChain is 45.94%.

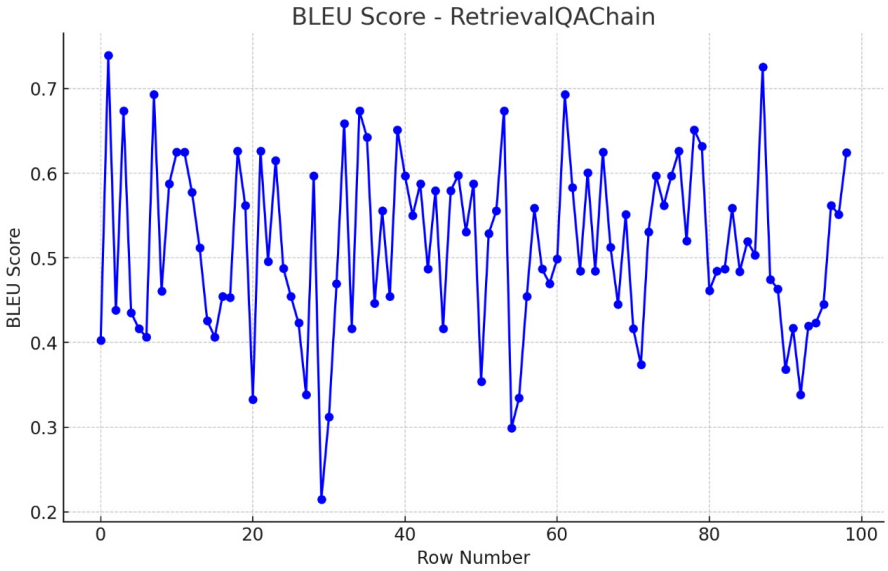

Figure 4.6: BLEU score of RetrievalAQChain recommendations

In Figure 4.6, the result shows that the accuracy we got for BLEU on RetrievalAQChain is 56.96%.

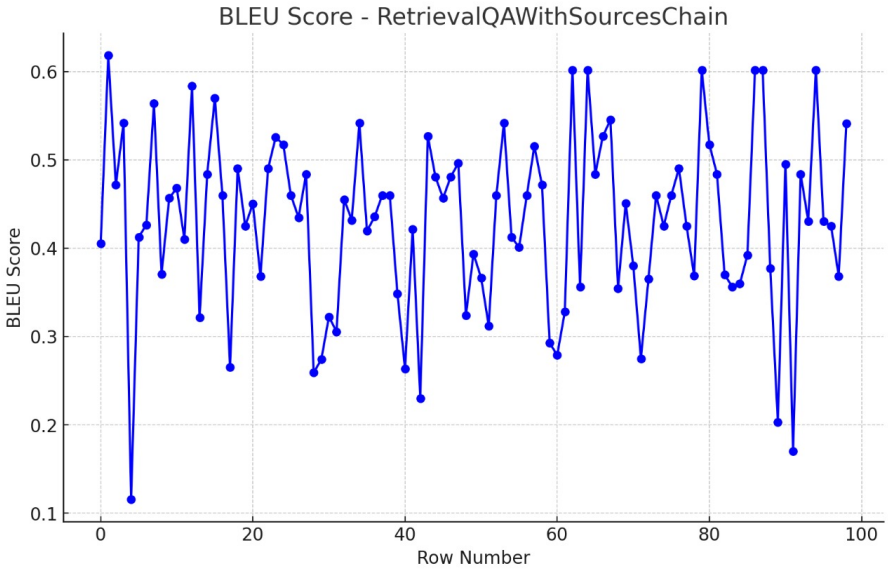

Figure 4.7: BLEU score of RetrievalAQWithSourcesChain recommendations

In Figure 4.7, the result shows that the accuracy we got for BLEU on RetrievalAQWith-SourcesChain is 42.61%.

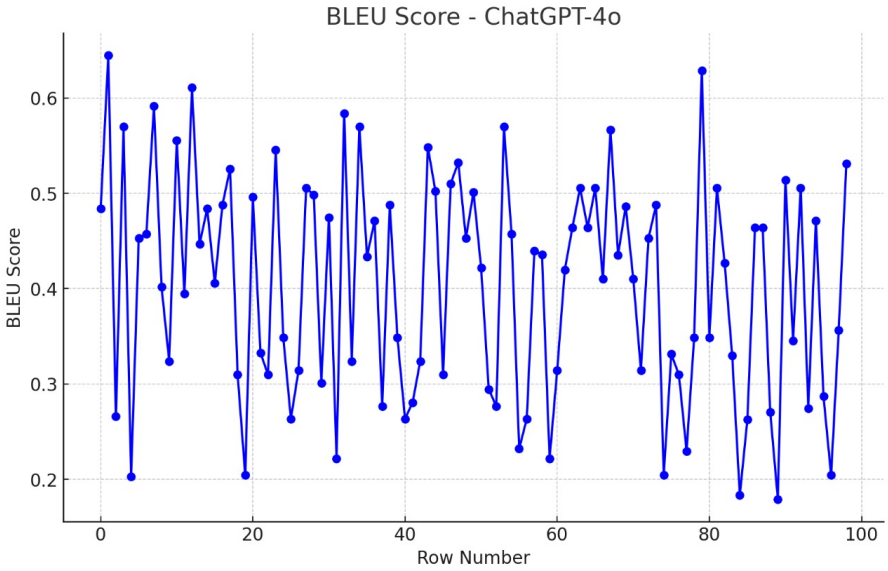

Figure 4.8: BLEU score of GPT-4 recommendations

In Figure 4.8, the result shows that the accuracy we got for BLEU on GPT-4 is 40.11%.

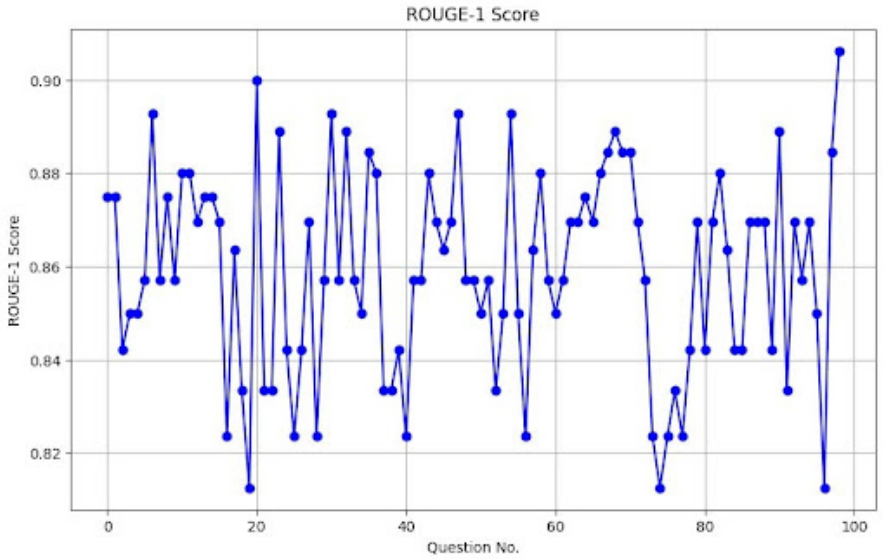

Figure 4.9: ROUGE-1 accuracy on GraphCypherQAChain

In Figure 4.9, the result shows that the accuracy we got for ROUGE-1 for GraphCypherQAChain is 86%.

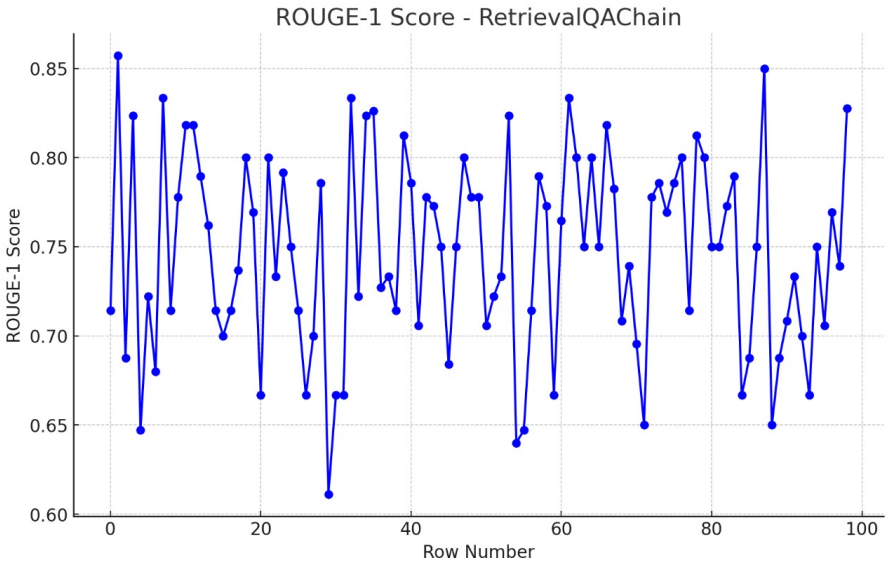

Figure 4.10: ROUGE-1 accuracy on RetrievalAQChain

In Figure 4.10, the result shows that the accuracy we got for ROUGE-1 on RetrievalAQChain is 74.54%.

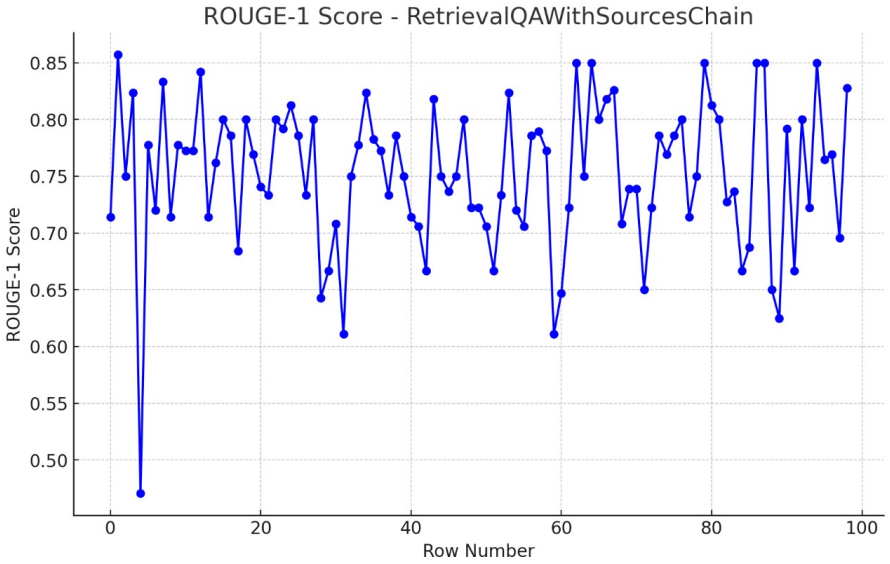

Figure 4.11: ROUGE-1 accuracy on RetrievalAQWithSourcesChain

In Figure 4.11, the result shows that the accuracy we got for ROUGE-1 on RetrievalAQWithSourcesChain is 74.55%.

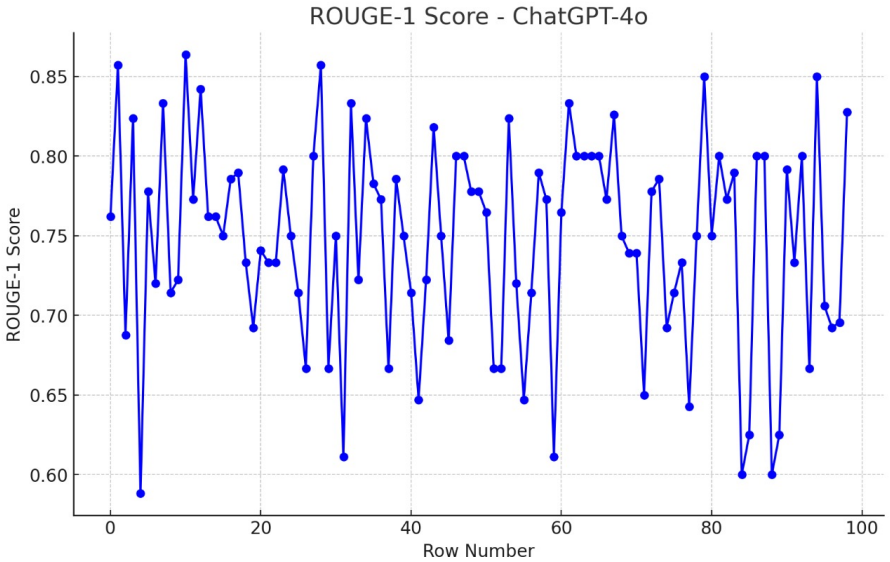

Figure 4.12: ROUGE-1 accuracy on GPT-4

In Figure 4.12, the result shows that the accuracy we got for ROUGE-1 on GPT-4 is 78.13%.

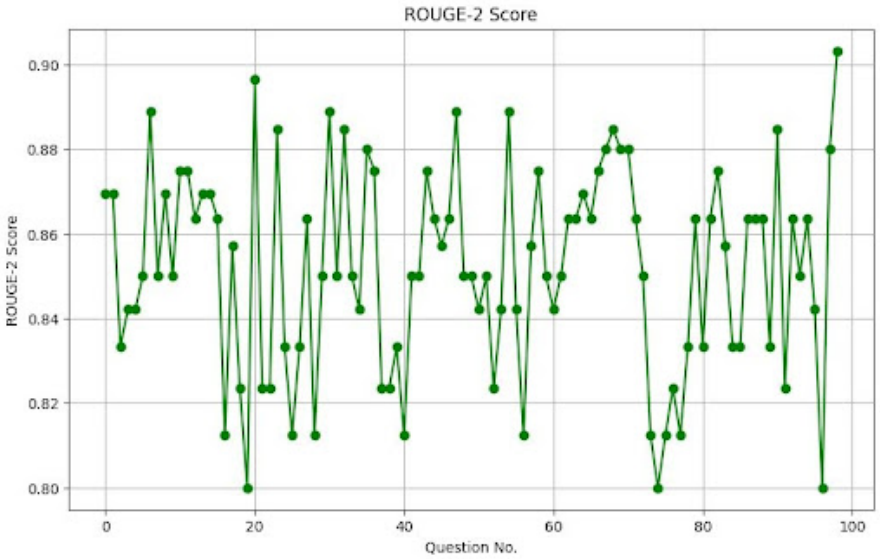

Figure 4.13: ROUGE-2 accuracy on GraphCypherQAChain

In Figure 4.13, the result shows that the ROUGE-2 accuracy we got on GraphCypherQAChain is 81.71%.

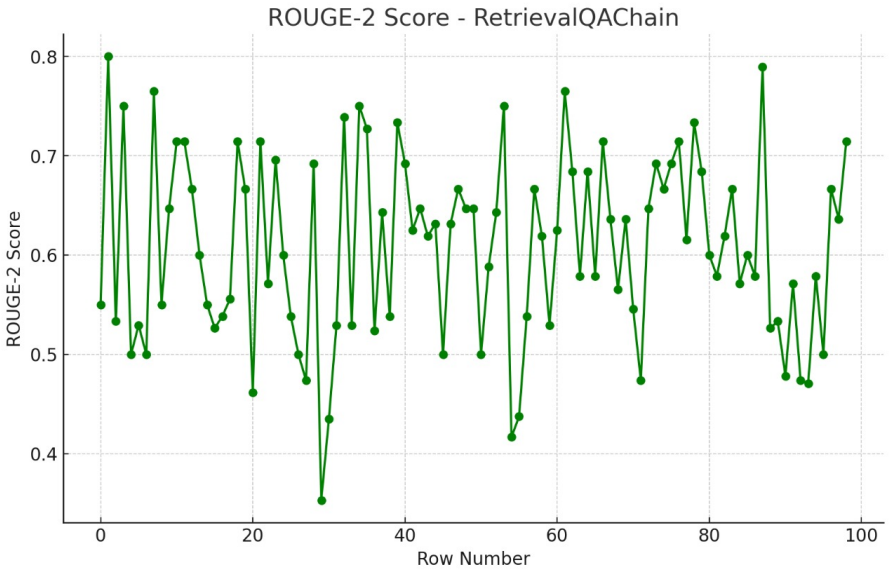

Figure 4.14: ROUGE-2 accuracy on RetrievalAQChain

In Figure 4.14, the result shows that the ROUGE-2 accuracy we got on RetrievalAQChain is 66.54%.

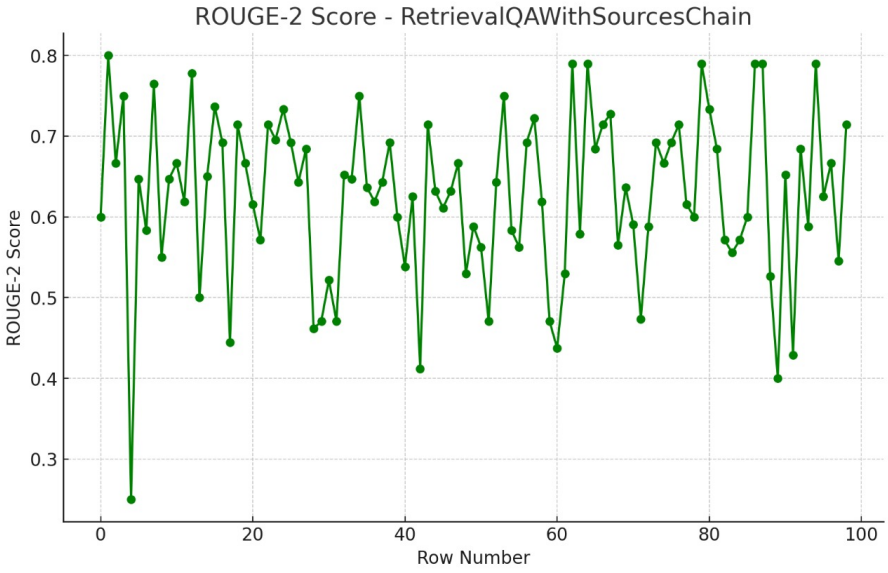

Figure 4.15: ROUGE-2 accuracy on RetrievalAQWithSourcesChain

In Figure 4.15, the result shows that the ROUGE-2 accuracy we got on RetrievalAQWith-SourcesChain is 66.10%.

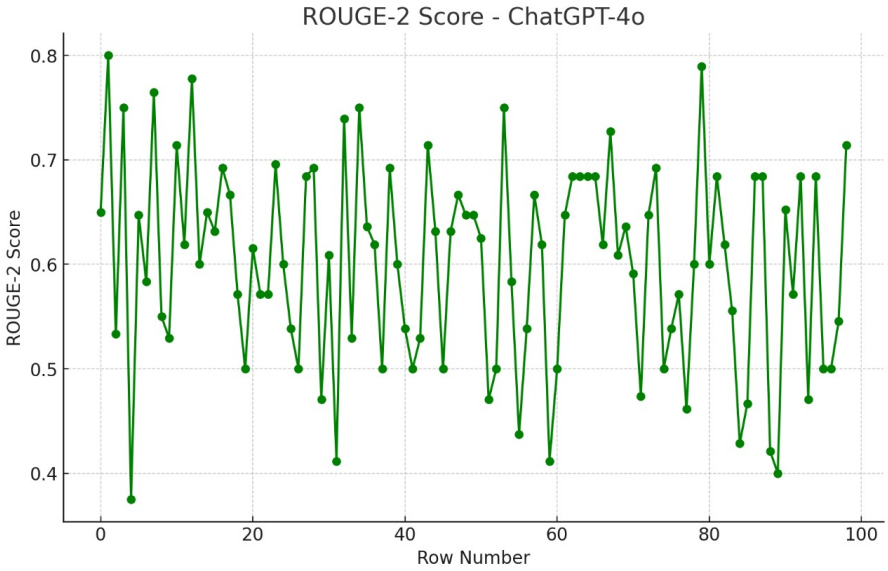

Figure 4.16: ROUGE-2 accuracy on GPT-4

In Figure 4.16, the result shows that the ROUGE-2 accuracy we got on GPT-4 is 60.32%.

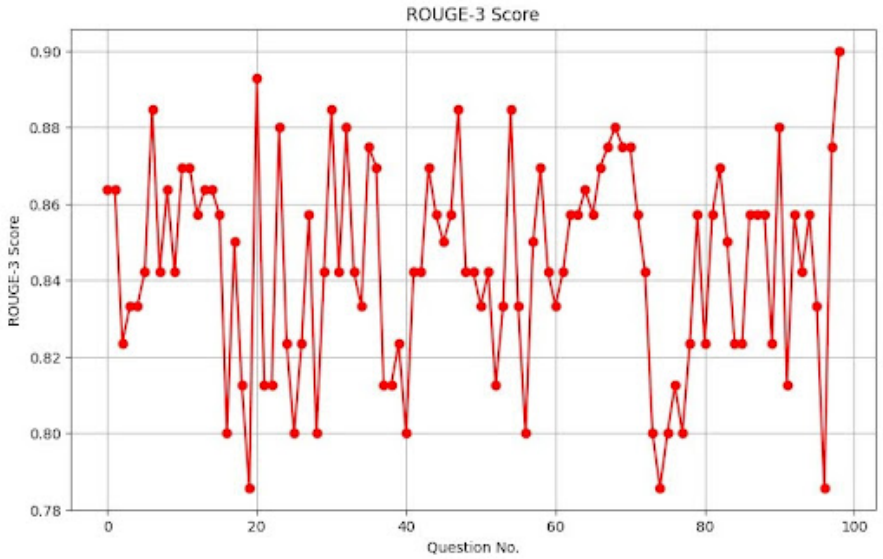

Figure 4.17: ROUGE-3 accuracy on GraphCypherQAChain

In Figure 4.17, the result shows that for GraphCypherQAChain the ROUGE-3 accuracy we got is 77.17%.

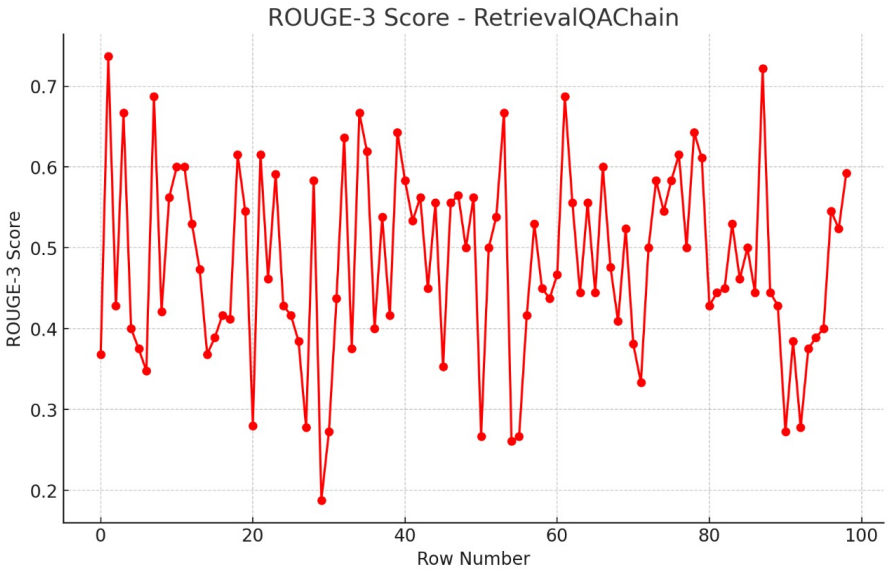

Figure 4.18: ROUGE-3 accuracy on RetrievalAQChain

In Figure 4.18, the result shows that for RetrievalAQChain the ROUGE-3 accuracy we got is 48.52%.

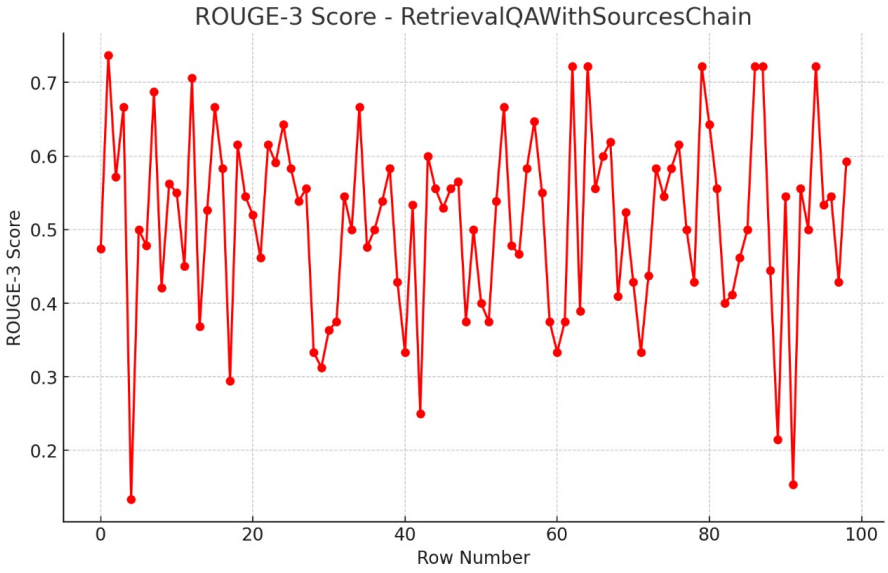

Figure 4.19: ROUGE-3 accuracy on RetrievalAQWithSourcesChain

In Figure 4.19, the result shows that for RetrievalAQWithSourcesChain the ROUGE-3 accuracy we got is 51.49%.

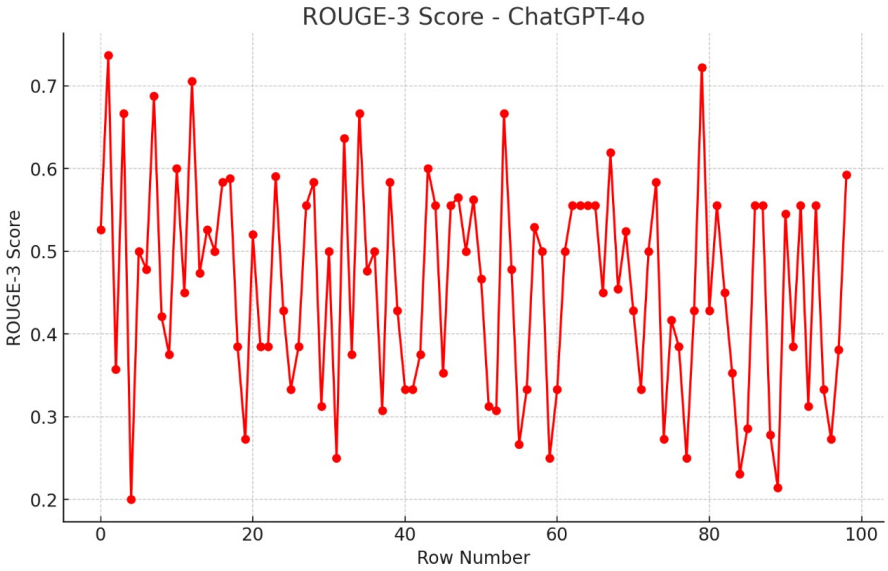

Figure 4.20: ROUGE-3 accuracy on GPT-4

In Figure 4.20, the result shows that for GPT-4 the ROUGE-3 accuracy we got is 48.13%.

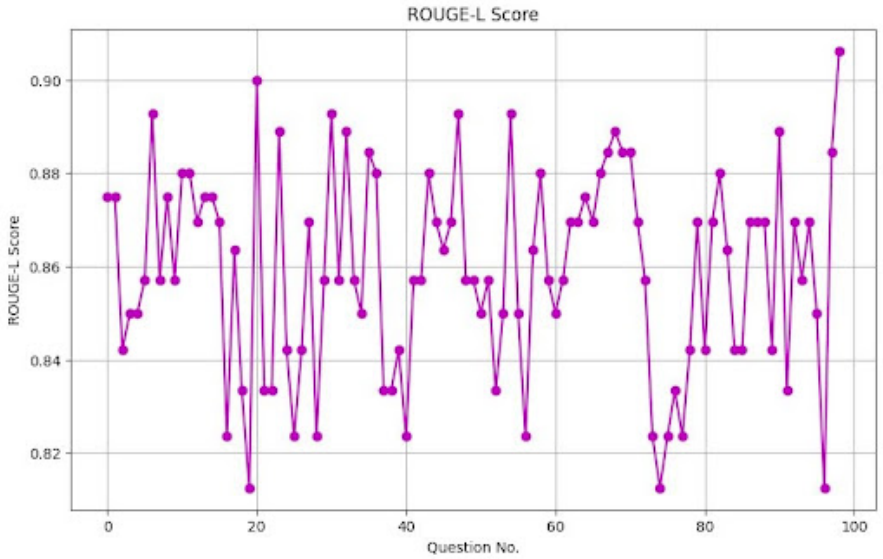

Figure 4.21: ROUGE-L accuracy on GraphCypherQAChain

In Figure 4.21, the result shows that ROUGE-L accuracy on GraphCypherQAChain is 86.26%.

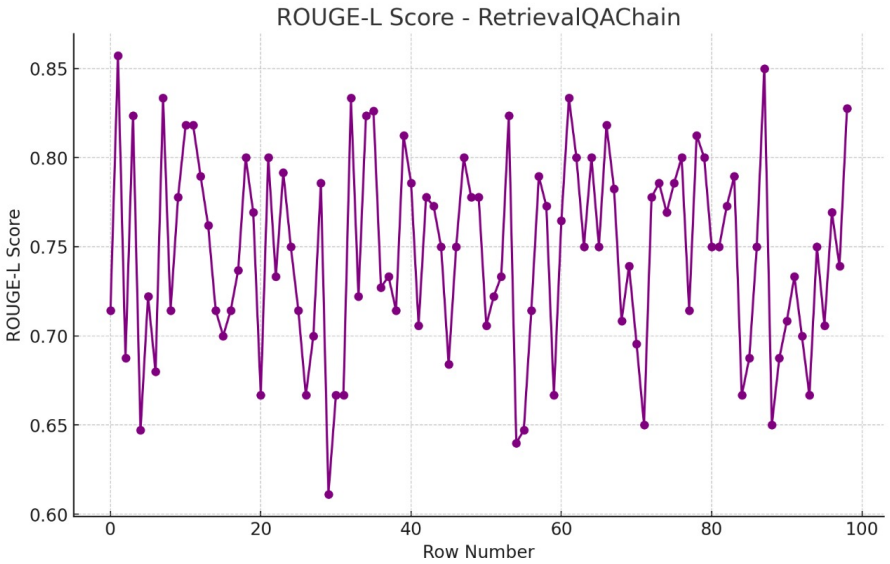

Figure 4.22: ROUGE-L accuracy on RetrievalAQChain

In Figure 4.22, the result shows that ROUGE-L accuracy on RetrievalAQChain is 64.64%.

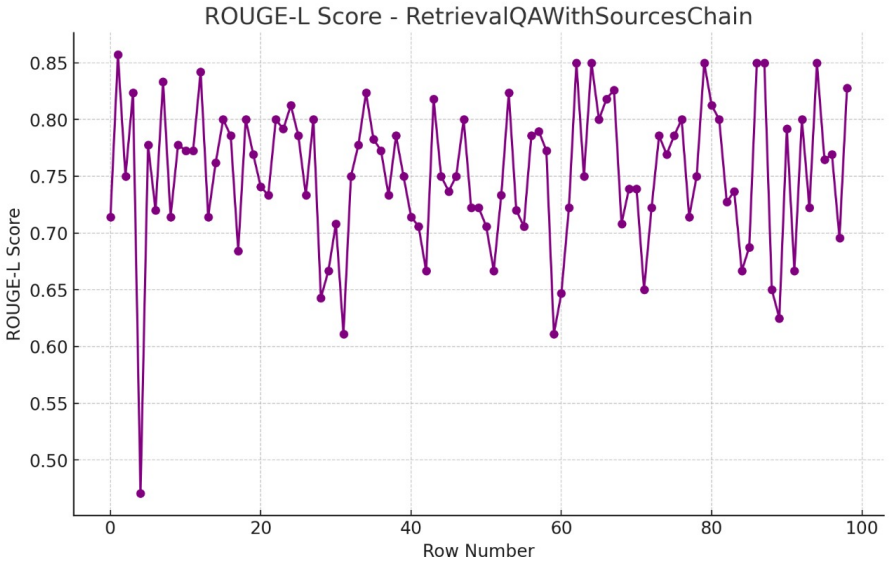

Figure 4.23: ROUGE-L accuracy on RetrievalAQWithSourcesChain

In Figure 4.23, the result shows that ROUGE-L accuracy on RetrievalAQWith-SourcesChain is 74.10%.

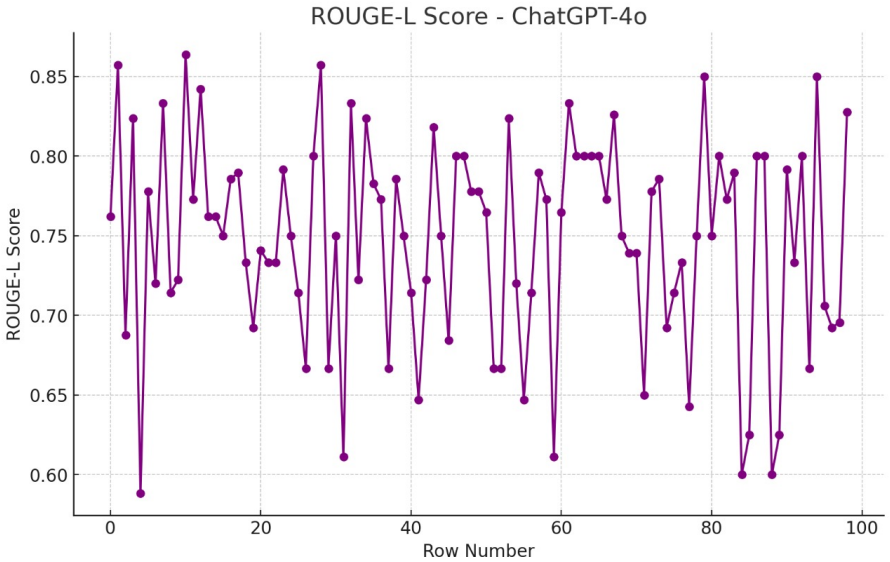

Figure 4.24: ROUGE-L accuracy on GPT-4

In Figure 4.24, the result shows that ROUGE-L accuracy on GPT-4 is 78.13%.

Table 4.1: 10 out of 100 diseases, their description and generated description of the Doctor Recommendation System using BLEU, ROUGE-1, ROUGE-2, ROUGE-3 and ROUGE-L

Disease	Description	Description_Generated
Acne	A skin condition involving blocked hair follicles and oil glands, leading to pimples, oily skin, and blackheads. Managed by a Dermatologist.	This is a condition characterized by a skin condition involving blocked hair follicles and oil glands, leading to pimples, oily skin, and blackheads. managed by a dermatologist.
Appendicitis	Inflammation of the appendix causing pain in the lower right abdomen, nausea, and fever. Requires surgery by a General Surgeon.	This is a condition characterized by inflammation of the appendix causing pain in the lower right abdomen, nausea, and fever. requires surgery by a general surgeon.
Cholesterol	High levels of cholesterol can lead to heart disease. Symptoms are often silent but may be associated with heart issues. Treated by a Cardiologist.	This is a condition characterized by high levels of cholesterol can lead to heart disease. symptoms are often silent but may be associated with heart issues. treated by a cardiologist.
Diabetes	A chronic condition affecting how the body processes blood sugar. Symptoms include frequent urination, excessive thirst, and fatigue. Treated by an Endocrinologist.	This is a condition characterized by a chronic condition affecting how the body processes blood sugar. symptoms include frequent urination, excessive thirst, and fatigue. treated by an endocrinologist.
Food Poisoning	Illness caused by consuming contaminated food, leading to vomiting, diarrhea, and abdominal cramps. Treated by a Gastroenterologist.	This is a condition characterized by illness caused by consuming contaminated food, leading to vomiting, diarrhea, and abdominal cramps. treated by a gastroenterologist.
GERD	A chronic condition where stomach acid flows back into the esophagus, causing heartburn and discomfort. Treated by a Gastroenterologist.	This is a condition characterized by a chronic condition where stomach acid flows back into the esophagus, causing heartburn and discomfort. treated by a gastroenterologist.

Heart Attack	A blockage in blood flow to the heart, causing chest pain, shortness of breath, and sweating. Treated by a Cardiologist.	This is a condition characterized by a blockage in blood flow to the heart, causing chest pain, shortness of breath, and sweating. treated by a cardiologist.
Jaundice	Yellowing of the skin and eyes due to high bilirubin levels, often caused by liver problems. Treated by a Hepatologist.	This is a condition characterized by yellowing of the skin and eyes due to high bilirubin levels, often caused by liver problems. treated by a hepatologist.
Migraine	A type of headache causing intense, throbbing pain, often accompanied by nausea and sensitivity to light. Treated by a Neurologist.	This is a condition characterized by a type of headache causing intense, throbbing pain, often accompanied by nausea and sensitivity to light. treated by a neurologist.
Rheumatoid Arthritis	A chronic autoimmune disorder that primarily affects the joints, causing inflammation and pain. Symptoms include joint pain, swelling, morning stiffness, and reduced range of motion. Treated by a Rheumatologist.	This is a condition characterized by a chronic autoimmune disorder that primarily affects the joints, causing inflammation and pain. symptoms include joint pain, swelling, morning stiffness, and reduced range of motion. treated by a rheumatologist.

Table 4.2: Accuracy scores of the Doctor Recommendation System using BLEU, ROUGE-1, ROUGE-2, ROUGE-3 and ROUGE-L

Disease	CypherQA-Chain	RetrievalQA-Chain	RetrievalQA-WithSources-Chain	GPT-4	Name
Acne	0.593941783	0.739195945	0.618233964	0.644600335	BLEU
	0.875	0.857142857	0.857142857	0.857142857	ROUGE-1
	0.869565217	0.8	0.8	0.8	ROUGE-2
	0.863636364	0.736842105	0.736842105	0.736842105	ROUGE-3
	0.875	0.857142857	0.857142857	0.857142857	ROUGE-L
Appendicitis	0.523171657	0.406445826	0.569786366	0.405698552	BLEU
	0.869565217	0.7	0.8	0.75	ROUGE-1
	0.863636364	0.526315789	0.736842105	0.631578947	ROUGE-2
	0.857142857	0.388888889	0.666666667	0.5	ROUGE-3
	0.869565217	0.7	0.8	0.75	ROUGE-L
Cholesterol	0.542858682	0.65859982	0.455120616	0.583899713	BLEU
	0.888888889	0.833333333	0.75	0.833333333	ROUGE-1
	0.884615385	0.739130435	0.652173913	0.739130435	ROUGE-2

	0.88	0.636363636	0.545454545	0.636363636	ROUGE-3
	0.888888889	0.833333333	0.75	0.833333333	ROUGE-L
Diabetes	0.506730989	0.486795502	0.526640388	0.548179946	BLEU
	0.88	0.772727273	0.818181818	0.818181818	ROUGE-1
	0.875	0.619047619	0.714285714	0.714285714	ROUGE-2
	0.869565217	0.45	0.6	0.6	ROUGE-3
	0.88	0.772727273	0.818181818	0.818181818	ROUGE-L
Food Poisoning	0.516461552	0.673604191	0.541728339	0.569598843	BLEU
	0.85	0.823529412	0.823529412	0.823529412	ROUGE-1
	0.842105263	0.75	0.75	0.75	ROUGE-2
	0.833333333	0.666666667	0.666666667	0.666666667	ROUGE-3
	0.85	0.823529412	0.823529412	0.823529412	ROUGE-L
GERD	0.558637449	0.55882652	0.515562573	0.439620385	BLEU
	0.863636364	0.789473684	0.789473684	0.789473684	ROUGE-1
	0.857142857	0.666666667	0.722222222	0.666666667	ROUGE-2
	0.85	0.529411765	0.647058824	0.529411765	ROUGE-3
	0.863636364	0.789473684	0.789473684	0.789473684	ROUGE-L
Heart Attack	0.577036236	0.583404106	0.601645611	0.464061161	BLEU
	0.869565217	0.8	0.85	0.8	ROUGE-1
	0.863636364	0.684210526	0.789473684	0.684210526	ROUGE-2
	0.857142857	0.555555556	0.722222222	0.555555556	ROUGE-3
	0.869565217	0.8	0.85	0.8	ROUGE-L
Jaundice	0.577036236	0.631749862	0.601645611	0.628450291	BLEU
	0.869565217	0.8	0.85	0.85	ROUGE-1
	0.863636364	0.684210526	0.789473684	0.789473684	ROUGE-2
	0.857142857	0.611111111	0.722222222	0.722222222	ROUGE-3
	0.869565217	0.8	0.85	0.85	ROUGE-L
Migraine	0.577036236	0.725433063	0.601645611	0.464061161	BLEU
	0.869565217	0.85	0.85	0.8	ROUGE-1
	0.863636364	0.789473684	0.789473684	0.684210526	ROUGE-2
	0.857142857	0.722222222	0.722222222	0.555555556	ROUGE-3
	0.869565217	0.85	0.85	0.8	ROUGE-L
Roughh-eumatoid Arthritis	0.61318085	0.62418867	0.540840984	0.530913078	BLEU
	0.90625	0.827586207	0.827586207	0.827586207	ROUGE-1
	0.903225806	0.714285714	0.714285714	0.714285714	ROUGE-2
	0.9	0.592592593	0.592592593	0.592592593	ROUGE-3
	0.90625	0.827586207	0.827586207	0.827586207	ROUGE-L

Table 4.1 is showing the generated descriptions and actual description for the diseases (10 out of 100 Disease) and Table 4.2 is showing their corresponding BLEU and ROUGE scores.

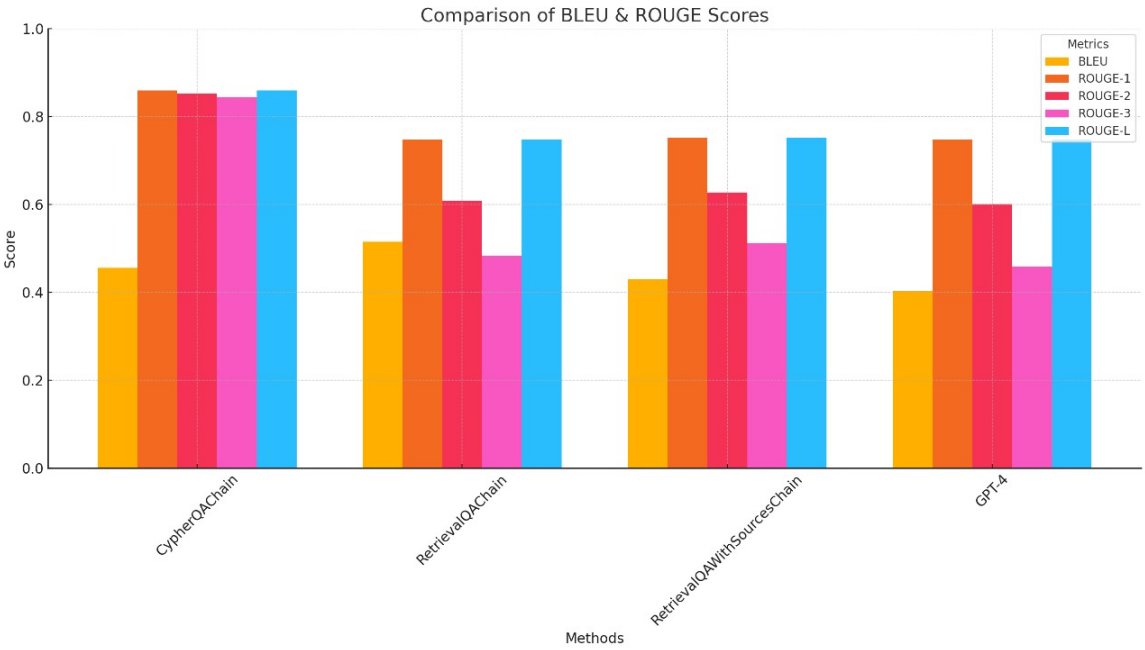

Figure 4.25: Average scores of CypherQAChain, RetrievalQAChain, RetrievalQAWithSourcesChain and GPT-4

Table 4.3: Overall accuracy score graph of CypherQAChain, RetrievalQAChain, RetrievalQAWithSourcesChain, GPT-4

Score	Proposed Chains for RAG Model			State of The Art
Name	CypherQA-Chain	RetrievalQA-Chain	RetrievalQA-WithSources-Chain	GPT-4
ROUGE-1	0.8626	0.7454	0.7455	0.7813
ROUGE-2	0.8171	0.6654	0.6610	0.6032
ROUGE-3	0.7717	0.4852	0.5149	0.4813
ROUGE-L	0.8626	0.6464	0.7410	0.7813
BLEU	0.4594	0.5696	0.4261	0.4011

Figure 4.25 is the visualization of Table 4.3 where it is evident that using both the Knowledge Graph and RAG together enhanced performance, even surpassed the accuracy of the latest GPT-4 across all evaluation metrics.

Table 4.4: Questions and answers of chains such as retrivalQA, graphCypherQA, retrivalQAWithSource

Chain Name	Question	Answer
retrivalQA	Tell me about the symptoms of diabetes and how to treat it	The symptoms of diabetes can vary depending on the type of diabetes. Common symptoms of diabetes may include increased thirst, frequent urination, extreme hunger, unexplained weight loss, fatigue, blurred vision, slow-healing sores, and frequent infections. To treat diabetes, it is important to work closely with a healthcare provider to develop a personalized treatment plan. Treatment typically involves a combination of healthy eating, regular physical activity, monitoring blood sugar levels, and taking medication as prescribed. For type 1 diabetes, insulin therapy is necessary. For type 2 diabetes, lifestyle changes such as diet and exercise are often the first line of treatment, along with oral medications and possibly insulin therapy. It is important for individuals with diabetes to regularly monitor their blood sugar levels, follow their treatment plan, and attend regular medical check-ups.
graphCypher-QA	Tell me about the symptoms of chicken pox and how to treat it.	The symptoms of Chicken pox include red spots over the body, malaise, swollen lymph nodes, mild fever, loss of appetite, headache, high fever, lethargy, fatigue, and skin rash. To treat Chicken pox, it is recommended to consult a healthcare professional for proper diagnosis and management, as well as to follow their prescribed treatment plan which may include antiviral medications, pain relievers, and soothing lotions to alleviate itching.
retrivalQA-WithSource	Tell me about the symptoms of diabetes and how to treat it and which doctor can treat it.	Symptoms of diabetes include increased thirst, frequent urination, unexplained weight loss, fatigue, blurred vision, and slow-healing sores. Diabetes can be treated by an Endocrinologist, who specializes in managing diseases like diabetes. Treatment typically involves lifestyle changes, medication (such as insulin or oral drugs), and monitoring blood sugar levels regularly.

Table 4.5: Questions and answers of Similarity Search

Question	Answer
Chicken pox	Chickenpox is a highly contagious disease caused by the varicella-zoster virus (VZV). It can cause an itchy, blister-like rash. The rash first appears on the chest, back, and face, and then spreads over the entire body, causing between 250 and 500 itchy blisters.

In the Table 4.4, we are getting relevant responses from our system for the queries. Also, from Similarity Search shown in Table 4.5, decent response is obtained even though it is not as powerful as chains.

Additionally, we focused on up-to-date data without training the model all again which we noticed other research papers had lacked about.

Table 4.6: Before updating database with the updated dataset

Question	Answer
Tell me about the symptoms of hypothyroidism and how to treat it.	I don't know the answer

Table 4.7: After updating database with the updated dataset

Question	Answer
Tell me about the symptoms of hypothyroidism and how to treat it.	The symptoms of hypothyroidism include abnormal menstruation, irritability, depression, swollen extremities, brittle nails, enlarged thyroid, puffy face and eyes, dizziness, lethargy, and mood swings. Treatment for hypothyroidism typically involves hormone replacement therapy, such as levothyroxine, to regulate thyroid hormone levels.

As shown in Table 4.6, the system initially refrained from providing ambiguous responses, indicating that it was unsure of the answer. However, after updating the dataset, the system demonstrated improved performance, as evidenced by the correct responses generated in Table 4.7. In the Retrieval-Augmented Generation (RAG) model, we avoid the time-consuming process of retraining the entire model on new data. Instead, we efficiently retrieve relevant information from our data source and apply the Large Language Model (LLM) to generate accurate responses. This approach not only accelerates the system's response time but also ensures that it remains up-to-date with the latest information.

4.3 Complexity Analysis

Table 4.8: Time Complexity of RAG model based Doctor Recommendation System

Component	Time Complexity	Description
Vector Indexing	$O(\log n)$	Approximate nearest-neighbor search to narrow down relevant nodes or documents.
Embedding Generation	$O(m)$	Time to generate embeddings (OpenAIEmbeddings) for user queries or documents.
Graph Query Execution	Traversal-based: $O(V_q + E_q)$ Indexed-based: $O(\log V)$	Traversal involves queried nodes V_q and edges E_q; indexed lookups are faster.
LLM Processing	$O(t^2)$	(For GPT-3.5-turbo) Processing complexity where t is the token count (query + retrieved results + prompt).
RetrievalQAChain	$O(q) + O(\log n) + O(t^2)$	Embedding query, vector search, and processing by GPT-3.5-turbo.
GraphCypherQAChain	$O(q) + O(V_q + E_q) + O(t^2)$ $O(q) + O(\log V) + O(t^2)$	Embedding query + graph traversal (or indexed lookup) + LLM processing.
RetrievalQAWithSourcesChain	$O(q) + O(\log n) + O(t^2)$	Similar to RetrievalQAChain but includes metadata extraction and source tracking..

where,

- V: Total number of nodes in the graph.
- V_q: Number of nodes retrieved based on the user's query.
- E_q: Number of relationships retrieved based on the user's query.
- n: Total number of documents or embeddings stored for vector search.
- t: Total token count processed by GPT-3.5-turbo (query + results).
- q: Number of tokens in the user query.
- m: Number of tokens processed during embedding generation.

From Table 4.8, the graph query execution uses index-based execution when the system needs to quickly find specific nodes. For example, a Doctor node with the type name 'Dermatologist' using pre-built indexes. Also, traversal-based execution happens when it needs to start from a node and explores relationships. For example, finding diseases treated by a 'Dermatologist' by navigating through the graph.

4.4 Discussion

The doctor recommendation system remains a critical component in healthcare applications. Through an extensive review of existing research papers, it became clear that many of these systems can become outdated over time [9], [11], [3], [7]. Additionally, relying solely on Large Language Models (LLMs) does not guarantee continuous updates. To address this challenge, we incorporated the RAG (Retrieval-Augmented Generation) model alongside a knowledge graph, which effectively mitigates this issue.

By evaluating various performance metrics—Correctness, Contextual Accuracy, CoT Contextual Accuracy, and Jaro-Winkler Distance using LangSmith—we found that the graphCypherQAChain outperformed the other two chains, retrievalQAChain and retrievalQAWithSourcesChain. Specifically, graphCypherQAChain achieved accuracy rates above 98%, 95%, and 97% across these metrics, respectively. Furthermore, in terms of distance, graphCypherQAChain demonstrated the smallest distance, around 22% lower than the other two chains.

To compare the performance of our model with GPT-4, we also used BLEU and ROUGE metrics, as we employed GPT-3.5-turbo in our approach. While GPT-3.5-turbo tends to be less accurate and sophisticated than GPT-4—particularly for complex tasks involving nuanced language, intricate reasoning, or deep contextual understanding—GPT-4 generally outperforms GPT-3.5-turbo in such areas. However, by integrating GPT-3.5-turbo with the knowledge graph and RAG model, specifically using graphCypherQAChain from the LangChain framework, we achieved noticeably better performance than GPT-4.

In terms of ROUGE and BLEU scores, our model attained the following results: 86% for ROUGE-1, 82% for ROUGE-2, 77% for ROUGE-3, 86% for ROUGE-L, and 46% for BLEU. In comparison, GPT-4 achieved 78%, 60%, 48%, 78%, and 40%, respectively. This improvement can be attributed to the careful focus on enhancing the retrieval process, rather than relying solely on the pre-trained LLM.

Additionally, training an LLM from scratch is both costly and time-consuming, requiring significant expertise and resources. Fine-tuning an existing model is more affordable and faster but still demands considerable effort. In this context, the graphCypherQAChain's retrieval mechanism proves beneficial. For example, as shown in Table 4.6, the "hypothyroidism" disease was initially unknown. However, after updating the knowledge graph with an updated dataset (Table 4.7), the system was able to provide accurate responses without needing to modify the LLM itself.

This highlights the effectiveness of combining knowledge graphs with retrieval-augmented models for continuous system improvement, reducing the need for extensive retraining and enabling more efficient adaptation to new information. Specifically, the contribution of graphCypherQAChain in this context is crucial, as it enhances the retrieval process by leveraging the knowledge graph's dynamic updating capabilities, allowing the system to integrate new data seamlessly without requiring modifications to the underlying LLM. This approach ensures that the system remains up-to-date and responsive to evolving information, offering a more scalable and cost-effective solution.

4.4.1 Findings Corresponding to the Research Questions

The first question explores how a knowledge graph can represent the relationships between doctors, diseases, and symptoms to support recommendations. The knowledge graph models doctors, diseases, and symptoms as nodes, with relationships such as "TREATS" linking doctors to diseases and "HAS_SYMPTOM" linking diseases to symptoms as shown in Figure 3.3. This interconnected undirected graph structure effectively captures the relationships, enabling the system to navigate and retrieve relevant information to support accurate recommendations. By leveraging these relationships, the system identifies diseases based on symptoms and, in turn, finds doctors who treat those diseases. Through Cypher queries, it dynamically traverses the graph to perform both direct and multi-step retrievals based on user input. Additionally, OpenAI embeddings enhance search accuracy by generating vectorized representations of nodes and their relationships which allows the system to leverage semantic proximity, especially for handling complex, multi-symptom queries.

Moving to the second question examines how the RAG model improves the accuracy and context-awareness of doctor recommendations. RAG retrieves information from external sources like a knowledge graph and uses it to generate responses through a language model such as GPT shown in Figure 2.2. This helps the model access the latest and most relevant data improving the accuracy of doctor recommendations based on symptoms and diseases. By pulling specific information for each query, RAG makes sure the recommendations fit the user's needs. Instead of relying only on internal data, RAG uses trusted sources to provide up-to-date and contextually relevant knowledge. This improves the accuracy of suggestions, especially when queries involve complex relationships between symptoms and treatments. RAG also reduces errors or "hallucinations" by grounding responses in verified data from sources like a knowledge graph or vector embeddings. This makes the recommendations more reliable as evident in Figure 4.1 and 4.4, as well as context-aware as evident in Figure 4.2, 4.3 and 4.25. Additionally, RAG can connect to different knowledge sources to provide information about doctors' specialties, diseases, their symptoms and who treats the diseases as shown in Table 4.1 which makes it a flexible and scalable solution for doctor recommendations.

Finally, the third question which is what strategies can ensure the recommendations remain up-to-date. Updating the knowledge graph with updated and relevant data helps include new relationships, doctors, diseases, and symptoms. This ensures that the RAG model retrieves the most current and relevant information for each query as evident in Table 4.6 and 4.7. By combining a language model (LLM) with a retrieval system, RAG can query the latest data from the knowledge graph or other external sources, ensuring that each recommendation reflects up-to-date knowledge. Additionally, incorporating user feedback into the doctor recommendation system is planned for future work. If a recommendation is not well-received, this feedback will be used to fine-tune the model for improving future recommendations.

Chapter 5

Conclusion

Graph representation learning have resulted in cutting edge achievements across various fields, including recommendation systems, question answering, and social network analysis. Thus, we presented an advanced Doctor Recommendation System that leverages the strengths of Knowledge Graphs and Large Language Model (LLM) to provide personalized and accurate doctor recommendations in this research. By utilizing Neo4j for constructing and querying a comprehensive Knowledge Graph, we effectively captured the complex relationships among various healthcare entities, including doctors, specialties, patient reviews, hospital affiliations, and medical conditions. This rich data foundation enabled the system to offer precise and contextually relevant recommendations.

The integration of a Retrieval-Augmented Generation (RAG) model using the LangChain framework significantly enhanced the system's ability to generate context aware suggestions. By combining retrieval-based and generation-based methods, our system dynamically provided updated responses tailored to individual patient queries and historical data, improving both accuracy and user satisfaction.

Experimental results demonstrated that the system performed well compared to the traditional recommendation systems in terms of accuracy and user experience, validating the efficacy of our approach for graphCypherQAChain. The successful implementation of this system highlights the potential of combining graph-based data organization with advanced natural language processing techniques to address complex recommendation challenges in the healthcare domain.

Future work will focus on expanding the scope of the Knowledge Graph to include more diverse data sources and further refining the RAG model for improved performance. Additionally, we aim to explore the integration of real-time patient feedback to continuously enhance the recommendation process. We also included fine-tuning in our future work, as fine-tuning is generally more expensive due to the need for extensive data, high computational requirements, and longer training times. This system sets a precedent for the development of sophisticated, data-driven recommendation systems that can significantly impact healthcare delivery and patient outcomes, paving the way for more personalized and efficient healthcare services.

Bibliography

[1] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2724–2743, 2017. DOI: 10.1109/TKDE.2017.2754499.

[2] S. Auer, V. Kovtun, M. Prinz, A. Kasprzik, M. Stocker, and M.-E. Vidal, "Towards a knowledge graph for science," Jun. 2018, pp. 1–6. DOI: 10.1145/3227609.3227689.

[3] M. Waqar, N. Majeed, H. Dawood, A. Daud, and N. R. Aljohani, "An adaptive doctor-recommender system," *Behaviour & Information Technology*, vol. 38, no. 9, pp. 959–973, 2019.

[4] D. Fensel, U. Şimşek, K. Angele, *et al.*, "Introduction: What is a knowledge graph?" *Knowledge graphs: Methodology, tools and selected use cases*, pp. 1–10, 2020.

[5] V. Karpukhin, B. Oğuz, S. Min, *et al.*, "Dense passage retrieval for open-domain question answering," *arXiv preprint arXiv:2004.04906*, 2020.

[6] P. Lewis, E. Perez, A. Piktus, *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[7] S. Mondal, A. Basu, and N. Mukherjee, "Building a trust-based doctor recommendation system on top of multilayer graph database," *Journal of Biomedical Informatics*, vol. 110, p. 103 549, Aug. 2020. DOI: 10.1016/j.jbi.2020.103549.

[8] J. Chicaiza and P. Valdiviezo-Diaz, "A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions," *Information*, vol. 12, no. 6, p. 232, 2021.

[9] P. Haque, S. Pranta, and S. Zoha, "Doctor recommendation based on patient syndrome using convolutional neural network," *EDU Journal of Computer and Electrical Engineering*, vol. 2, pp. 30–36, Dec. 2021. DOI: 10.46603/ejcee.v2i1.36.

[10] S. Ji, S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 2, pp. 494–514, 2021.

[11] C. Ju and S. Zhang, "Doctor recommendation model based on ontology characteristics and disease text mining perspective," *BioMed Research International*, vol. 2021, no. 1, p. 7 431 199, 2021.

[12] Q. Guo, F. Zhuang, C. Qin, *et al.*, "A survey on knowledge graph-based recommender systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3549–3568, 2022. DOI: 10.1109/TKDE.2020.3028705.

[13] T. Tang, J. Li, W. X. Zhao, and J.-R. Wen, "Context-tuning: Learning contextualized prompts for natural language generation," *arXiv preprint arXiv:2201.08670*, 2022.

[14] AmberSearch. "What is retrieval-augmented generation?" Accessed: 2024-11-23. (2023), [Online]. Available: https://ambersearch.de/what-is-retrieval-augmented-generation/.

[15] L. Chen, M. Zaharia, and J. Zou, "How is chatgpt's behavior changing over time?" *arXiv preprint arXiv:2307.09009*, 2023.

[16] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "Ragas: Automated evaluation of retrieval augmented generation," *arXiv preprint arXiv:2309.15217*, 2023.

[17] Y. Gao, Y. Xiong, X. Gao, *et al.*, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, 2023.

[18] C. Jeong, "Generative ai service implementation using llm application architecture: Based on rag model and langchain framework," *Journal of Intelligence and Information Systems*, vol. 19, pp. 129–164, Dec. 2023. DOI: 10.13088/jiis.2023.29.4.129.

[19] J. Liu, C. Liu, R. Lv, K. Zhou, and Y. B. Zhang, "Is chatgpt a good recommender? a preliminary study," *ArXiv*, vol. abs/2304.10149, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258236609.

[20] K. Martineau, *Rag is an ai framework for retrieving facts from an external knowledge base to ground large language models (llms) on the most accurate, up-to-date information and to give users insight into llms' generative process.* Aug. 2023. [Online]. Available: https://research.ibm.com/blog/retrieval-augmented-generation-RAG?ref=blog.getdataring.com.

[21] K. Pichai, "A retrieval-augmented generation based large language model benchmarked on a novel dataset," *Journal of Student Research*, vol. 12, no. 4, 2023.

[22] S. Siriwardhana, R. Weerasekera, E. Wen, T. Kaluarachchi, R. Rana, and S. Nanayakkara, "Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 1–17, 2023. DOI: 10.1162/tacl_a_00530. [Online]. Available: https://aclanthology.org/2023.tacl-1.1.

[23] A. Wang, L. Song, G. Xu, and J. Su, "Domain adaptation for conversational query production with the rag model feedback," Jan. 2023, pp. 9129–9141. DOI: 10.18653/v1/2023.findings-emnlp.612.

[24] S. Chen, C. Wen, and L. Jin, "Design and implementation of medical intelligent questions answering system based on knowledge graph," in *2024 IEEE 7th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, IEEE, vol. 7, 2024, pp. 1814–1818.

[25] M. Jin, Q. Yu, D. Shu, *et al.*, *Health-llm: Personalized retrieval-augmented disease prediction system*, 2024. arXiv: 2402.00746 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2402.00746.

[26] Q. Liu, Y. Jin, X. Cao, *et al.*, "An entity ontology-based knowledge graph embedding approach to news credibility assessment," *IEEE Transactions on Computational Social Systems*, 2024.

[27] J. C. L. Ong, L. Jin, K. Elangovan, *et al.*, *Development and testing of a novel large language model-based clinical decision support systems for medication safety in 12 clinical specialties*, 2024. arXiv: 2402.01741 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2402.01741.

[28] K. Rajkumar, T. Ragupathi, and S. Karthikeyan, "Intelligent chatbot for hospital recommendation system," in *2024 2nd International Conference on Disruptive Technologies (ICDT)*, IEEE, 2024, pp. 664–668.

[29] K. Soman, P. W. Rose, J. H. Morris, *et al.*, "Biomedical knowledge graph-optimized prompt generation for large language models," *Bioinformatics*, vol. 40, no. 9, btae560, 2024.

[30] K. D. Spurlock, C. Acun, E. Saka, and O. Nasraoui, "Chatgpt for conversational recommendation: Refining recommendations by reprompting with feedback," *arXiv preprint arXiv:2401.03605*, 2024.

[31] J. Wu, J. Zhu, Y. Qi, *et al.*, *Medical graph rag: Towards safe medical large language model via graph retrieval-augmented generation*, 2024. arXiv: 2408.04187 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2408.04187.

[32] L. Xu, L. Lu, M. Liu, C. Song, and L. Wu, "Nanjing yunjin intelligent question-answering system based on knowledge graphs and retrieval augmented generation technology," *Heritage Science*, vol. 12, no. 1, p. 118, 2024.

[33] Z. Zhao, W. Fan, J. Li, *et al.*, "Recommender systems in the era of large language models (llms)," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[34] E. ELGAZAR, *Doctor's Specialty Recommendation*, Kaggle, Accessed: Jun 19, 2023. [Online]. Available: https://www.kaggle.com/datasets/ebrahimelgazar/doctor-specialist-recommendation-system.

[35] P. Eranga, *Disease prediction based on symptoms*, Kaggle, Accessed: Jun 8, 2023. [Online]. Available: https://www.kaggle.com/datasets/pasindueranga/disease-prediction-based-on-symptoms.