AI-DRIVEN KNOWLEDGE GRAPH FOR HEALTHCARE INSIGHTS

Roshni Venkat

Presentation: <https://github.com/roshniven/AI_in_healthcare>

Code: <https://github.com/roshniven/AI_in_healthcare>

Video:

<https://utexas.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=734aac40-fdef-40ef-83be-b2c1000df6cc>

**1. Introduction**

A large portion of healthcare data is unstructured and highly complex, making it difficult to process and analyze. Developers spend a lot of time working with multiple files of tabular data, writing complicated queries to extract useful information. This process is not only time-consuming but also requires significant computing power to repeatedly generate additional data sources. To solve these problems, my project experiments with using a knowledge graph to store and organize healthcare data. This allows data to be stored as a structured graph, helping us query and uncover new insights without extra processing. In addition, I make use of LangChain and OpenAI’s large language models (LLMs) to efficiently query the database using natural language. This eliminates the need to write and test complex queries and makes the database accessible to everyone, from healthcare practitioners to researchers. Finally, I experiment with creating a Graph Data Science (GDS) graph to cluster patients and identify patterns in their conditions. By implementing these techniques in this project, I aim to restructure the way data is aggregated, stored, and queried to enhance performance and extract valuable insights from healthcare data.

**2. Related Work**

Several studies have explored knowledge graph-based healthcare analytics. Xiang et al. [1] conducted a comprehensive survey on knowledge graph-based clinical decision support systems, emphasizing their role in structuring medical data for better reasoning and diagnosis. Their study outlined various reasoning mechanisms that enhance decision-making and improve the interpretability of medical recommendations. This aligns with my approach of using a knowledge graph to structure patient medical records, allowing for effective querying and analysis.

Similarly, Rotmensch et al. [2] developed a health knowledge graph extracted from electronic medical records (EHRs) to predict medical conditions. Their system demonstrated the potential of using machine learning models to analyze structured relationships in EHRs, improving diagnostic accuracy. Zhou et al. [3] introduced a human symptom-disease network, analyzing disease comorbidities through large-scale medical data. Their study demonstrated how network-based methods can reveal hidden patterns in disease progression. My project extends this by applying graph-based analytics such as community detection and mortality rate analysis to uncover trends in patient populations and identify high-risk individuals.

**3. Methodology**

My approach follows a multi-step workflow (see figure 1 below) to build, query, and analyze the healthcare knowledge graph.

A diagram of a graph

AI-generated content may be incorrect.

Figure 1: Solution workflow

**3.1 Data Processing and Knowledge Graph Construction**

The dataset consists of COVID-19 synthetic patient data and the following tables are used: *patients.csv*, *conditions.csv*, *encounters.csv*, and *medications.csv*. I initially designed the data schema using their managed cloud service, AuraDB, and saved it as an import script in the Cypher query language. I then executed this script on a local instance, which I set up using Neo4j Desktop, by installing and configuring a new database environment. Running the script imported data directly from the CSV files into the graph, where each table (such as patients and medicines) was represented as a node, with arrows indicating the relationships between entities. This structure is illustrated in Figure 2.

**A diagram of a patient flow

AI-generated content may be incorrect.**

Figure 2: Relationships between entities in the graph

### **3.2 AI-Powered Querying and Response Generation**

To enable natural language interactions, I integrated LangChain with OpenAI’s GPT-4 and GPT-3.5. Both of these models are used in the cypher chain to query data from the knowledge graph as well as convert the results of the query into a natural language response. Figure 1 illustrates the entire process. The process starts with the user inputting a natural language question. GPT-4 interprets this query and generates a corresponding Cypher query (see figure 3 for an example of this). With the extensive data encoded into the knowledge graph, the primary challenge is ensuring the accurate and efficient retrieval of information. Converting a natural language question into a precise Cypher query is a complex task, and in my implementation, GPT-4 proved to be the most effective model for this purpose. Once the Cypher query is executed, the retrieved data is structured in a JSON format, which is then passed to GPT-3.5 for final processing.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 3: Example of a Cypher query generated by GPT-4

The role of GPT-3.5 in this pipeline is to analyze the extracted data and construct a well-formulated textual response based on the query results. While the structured JSON data retrieved from the database may contain redundant or excessive information, GPT-3.5 efficiently filters and processes the necessary details to generate a contextually relevant response. This response is then printed to the user as output (see figure 4 for an example of this). This system ensures that even users unfamiliar with Cypher can retrieve meaningful insights from the healthcare knowledge graph. Additionally, using two LLMs in this process allows for a seamless transition from natural language input to structured query execution and an easy-to-understand response.

A close-up of a prescription

AI-generated content may be incorrect.

Figure 4: Example of a natural language response generated by GPT-3.5

### **3.3 Graph Analytics: Patient Clusters and Mortality Rates**

Using the Neo4j Graph Data Science (GDS), I created an undirected graph of patients and conditions. I ran the Louvain algorithm on this graph to create clusters (communities) of patients that have similar conditions. In addition, I created several queries to find the most prominent conditions in patient clusters as well as patients that were the most influential in spreading symptoms. I ran the cypher chain on the natural language questions for this graph, but GPT-4 was not able to generate correct Cypher queries for my questions, using incorrect features from the dataset. Thus, I resorted to using my own queries to retrieve patient data as well as calculate mortality rates.

### **4 Results**

I tested the LLM with various tasks that involved summarization, reasoning, extracting, filtering, and processing of the healthcare dataset. For most of my queries, the solution provided accurate and concise responses. However, its key vulnerability was generating Cypher queries for my prompts: it didn’t understand the crux of the question and provided an unrelated response. This was especially highlighted when I asked it to give me statistics for the clusters of patients and calculate mortality rates. If it understood the prompt, it did a phenomenal job at processing and retrieving data. The responses to my queries were generated in milliseconds, which illustrates the potential and magnitude of AI-driven data analytics.

### **5 Conclusion**

In this project, I developed an AI-driven knowledge graph to improve the accessibility and analysis of healthcare data. Using a Neo4j knowledge graph, I structured and stored multi-file tabular data, making it more organized and easier to retrieve. I integrated LangChain with OpenAI’s GPT-4 and GPT-3.5 to convert natural language questions into Cypher queries, allowing anyone to access medical data without requiring database expertise. Additionally, I created an undirected graph using Neo4j’s Graph Data Science tools to identify patient clusters and discover valuable insights into their medical conditions.

While the system worked well for most queries, there are several improvements that can be made. We can improve Cypher query generation through additional training and reinforcement learning. We can expand the system’s domain knowledge by integrating external datasets, such as clinical trials and drug interactions, to provide more comprehensive insights. Additionally, we can train and test with additional data sources including real-world healthcare data to improve reliability. In fact, we can train large language models (LLMs) on unstructured data like doctor’s notes and patient reports to uncover deeper insights. Finally, we can implement explainable AI methods to enhance transparency, making the model’s decisions easier to understand and increasing our trust in its outputs.

**6. References**

## [1] Xiang, X., Wang, Z., Jia, Y., & Fang, B. "Knowledge Graph-Based Clinical Decision Support System Reasoning: A Survey." IEEE.

[2] Rotmensch, M., Halpern, Y., Tlimat, A., Horng, S., & Sontag, D. (2017). "Learning a Health Knowledge Graph from Electronic Medical Records". Scientific Reports.

[3] Zhou, X., Menche, J., Barabási, A.-L., & Sharma, A. (2014). "Human Symptoms-Disease Network". Proceedings of the National Academy of Sciences.