2024

Capstone Final Report

Medical Chatbot for Enhanced Patient Data Insights

Ron Hankey

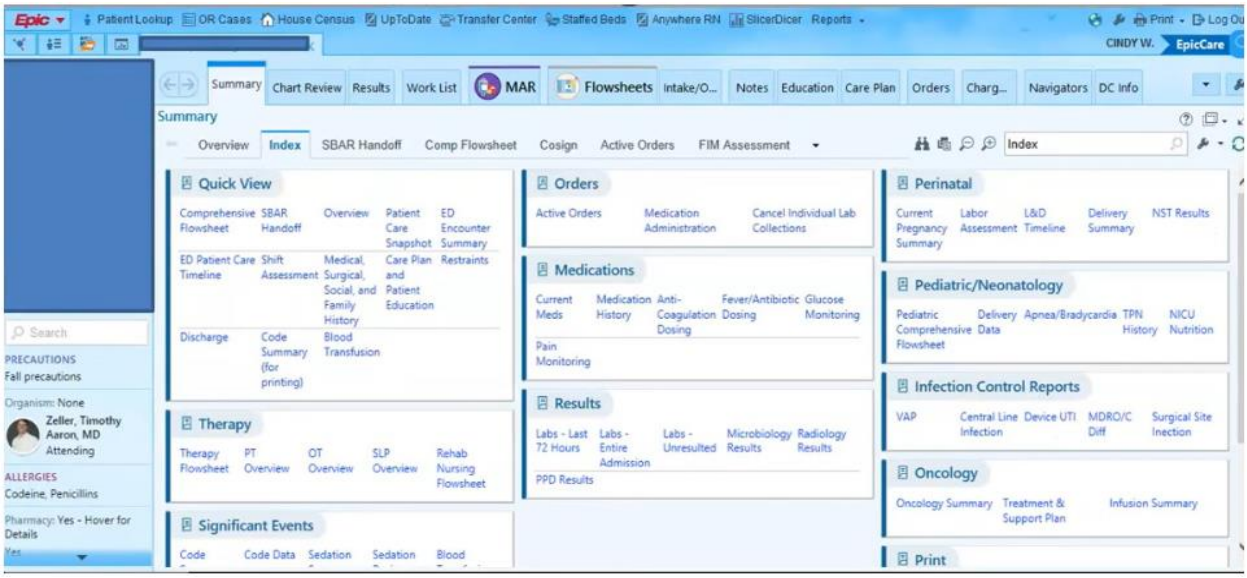
Springboard Data Science Cohort – Apr 2024

4/10/2024

Table of Contents

Problem identification

Physicians, nurses, hospital administration and other patient caregivers need to review patient records in electronic health records (EHR). As the screen image from Epic, one of the most popular EHR's, there is an enormous amount of data on just the summary page. Then there are multiple tabs with additional information that is important for patient care.



The amount of time it just takes a care giver to review a patient chart has been studied and one paper published stated this:

"The amount of time that providers spend using electronic health records (EHRs) to support the care delivery process is a concern for the U.S. health care system. Given the potential effect on patient care and the high costs related to this time, particularly for medical specialists whose work is largely cognitive, these findings warrant more precise documentation of the time physicians invest in these clinically focused EHR functions. … **Physicians spent an average of 16 minutes and 14 seconds per encounter using EHRs, with chart review (33%)**,

documentation (24%), and ordering (17%) … The proportion of time spent on various clinically focused functions was similar across specialties." (Physician Time Spent Using the Electronic Health Record During Outpatient Encounters: A Descriptive Study)

The deliverable for this project is a Generative AI (GenAI) model utilizing retrieval augmented generation (RAG) to leverage Large Language Models (LLM's) in an attempt to reduce the amount of time physicians, nurses and other caregivers spend in chart review.

Data wrangling

The project needed data that included a combination of unstructured text for natural language processing (NLP) and structured data for database querying. The unstructured data is needed in order to show the power of the LLM and the structured data was needed to show how a LLM could build the SQL to query the data base.

Medical data consisting of patient records is extremely hard to find due to Medical data is extremely hard to find due to Health Insurance Portability and Accountability Act (HIPAA) privacy regulations. According the CDC's website: "The Health Insurance Portability and Accountability Act of 1996 (HIPAA) is a federal law that **required the creation of national standards to protect sensitive patient health information from being disclosed without the patient's consent or knowledge**."

There were two data sets found on Kaggle that, while they were not ideal, they Offered a partial solution by combining the two data sets into one. These data sets were the Medical Records Dataset, this dataset contains simulated medical records for a fictional group of patients. The dataset was generated using the Python Faker library to create realistic but fake data (Medical Records Dataset).

The data set is described on Kaggle as having these columns:

Patient ID: A unique identifier for each patient (integer).

Name: A randomly generated full name (string).

Date of birth: A randomly generated date of birth with ages between 1 and 100 years old (date).

Gender: A randomly selected gender (M or F) (string).

Medical conditions: **A list of three random, unique words representing medical**

conditions (string).

Medications: **A list of three random, unique words representing medications (string).**

Allergies: **A list of three random, unique words representing allergies (string).**

Last appointment date: A randomly generated date within the range of the last 2 years (date).

The random words in the data set were not medical related, they were simply random words pulled from a dictionary so were not realistic in a medical sense.

There was still a need to a data set that provided unstructured text that corresponded to medical notes. That dataset also, found on Kaggle was named Medical Transcriptions. This dataset is described as follows on Kaggle: “Medical data is extremely hard to find due to HIPAA privacy regulations. This dataset offers a solution by providing medical transcription samples.” The dataset was de-identified so there was nothing to link any of the records to a specific patient. The data set consisted of the following columns:

description (Short description of transcription) (String)

medical_specialty (Medical specialty classification of transcription) (String)

sample_name (Transcription title) (String)

transcription (Sample medical transcriptions) (String)

keywords (Relevant keywords from transcription) (String)

Merging these two data sets would create a dataset for the needs of the generative AI model that was built for this project. From the Medical Records Dataset, the following columns were kept:

Patient ID: A unique identifier for each patient (integer).

Name: A randomly generated full name (string).

Date of birth: A randomly generated date of birth with ages between 1 and 100 years old (date).

Gender: A randomly selected gender (M or F) (string).

From the Medical Transcriptions dataset all of the columns were kept and a new medications and allergies column were added that contained no data from the Medical Records Dataset.

The next steps to wrangle data were:

1) Randomly assign patients from the Medical Records Dataset to rows in the Medical Transcriptions based on descriptions in the transcription. This was done to align medical notes with the correct gender (M, F).
2) Next the first names also had to be adjusted to align with gender.
3)
4)

Exploratory Data Analysis

The data was examined for concepts that could test the generative AI model. This was done by examining the medical concepts in the transcription column and the columns that would require calculation. The result was a series of questions that would be used to the model. The questions were as follows:

Which patients use cigarettes?

Which patients are over 50 years of age?

Which patients take percocet, search columns TRANSCRIPTION and MEDICATIONS?

Which patients have had a colonoscopy? Include patient ID and name?

Who is the oldest patient and how old are they?

Who is the youngest patient and what is their age?

Can you write one page with recommendations for a patient to help them lose weight.

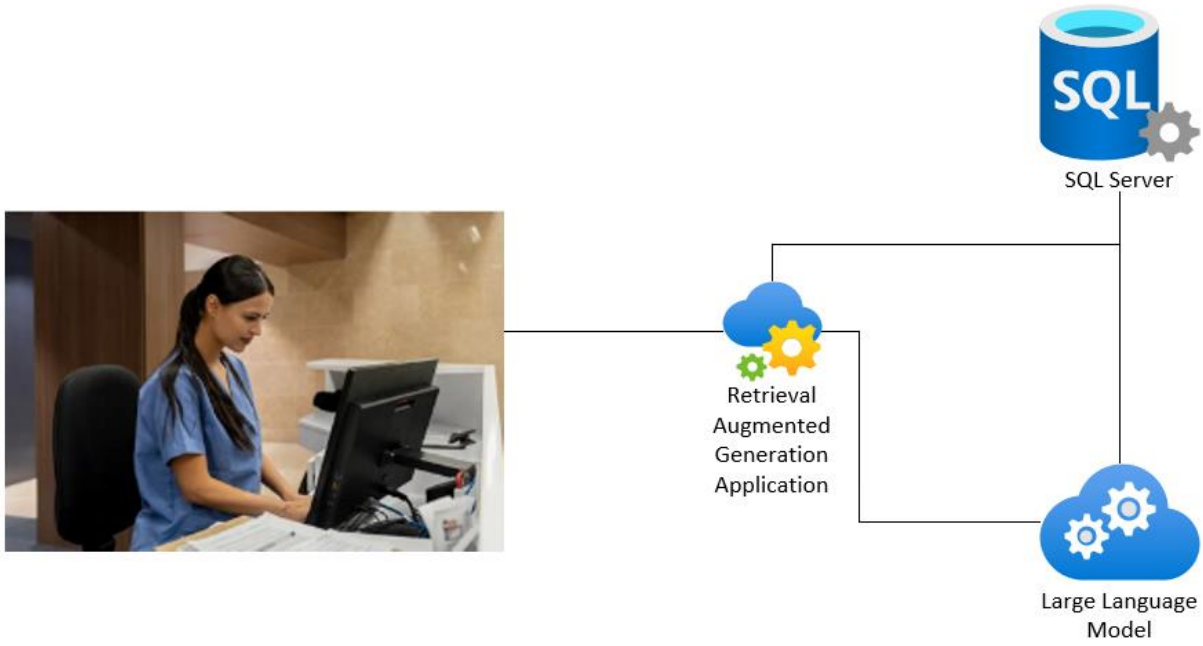
Write a one page document describing how to lower blood sugar naturally.

List all the patient ID's and names in the database that have appointments today.

List all the patient ID's and names in the database that have appointments tomorrow.

A series of SQL statements were created that would then be used in testing to verify that generative AI model was producing the correct answer.

Modeling



- **Generative AI (GenAI)**
  - A type of AI that can create new content and ideas, including conversations, stories, images, videos, and music
- **Large Language Model (LLM)**
  - Type of artificial intelligence (AI) program that can recognize and generate text, among other tasks.
- **Retrieval Augmented Generation (RAG)**
  - The process of optimizing the output of a large language model, so it references an authoritative knowledge base outside of its training data sources before generating response.
- **Langchain**
  - LLM orchestration framework that helps developers build generative AI applications or retrieval-augmented generation

(RAG) workflows.

- **Streamlit**
  - Open-source Python framework for data scientists and AI/ML engineers to deliver dynamic data apps with only a few lines of code
- **Database**
  - Microsoft SQL Server but any relational database would work

Documentation

Summary