**Introduction of using DoctorAI and Ultrasound Education Assistant**

**Document created for LLM and BIO Hackathon**

**Dr. JEFF XU ZHE**

**20JUL2023**

1. **DoctorAI**

Let's go through the key components of DoctorAI:

1. **Medical Knowledge Source: NHS.UK** NHS.UK is the primary medical knowledge source for DoctorAI. It contains valuable information about various medical conditions, symptoms, treatments, and other healthcare-related topics. The creator of DoctorAI has reviewed and controlled the data from NHS.UK to ensure its accuracy and reliability.
2. **Data Scraping Program** To collect the medical knowledge from NHS.UK, a data scraping program has been implemented. This program extracts relevant information from the NHS.UK website, such as articles, guidelines, and frequently asked questions, and stores it in a structured format for further processing.
3. **Embedding and Vector DB Creation Program** The data obtained from the data scraping program needs to be transformed into a format suitable for natural language processing (NLP) tasks. The Embedding and Vector Database Creation Program is responsible for converting the text data into numerical embeddings, which can be used for similarity matching and other NLP tasks. These embeddings are then stored in a database for efficient retrieval during runtime.
4. **Main Telegram Bot and QnA Program** The core of DoctorAI is the Telegram bot, which serves as the user interface. It receives queries from users, processes their input, and provides medical diagnoses and recommendations based on the knowledge collected from NHS.UK. The QnA (Question and Answer) program handles user queries, performs semantic matching with the pre-processed data, and retrieves relevant responses.
5. **Files Used to Create the Docker for Google Cloud Run** Google Cloud Run allows applications to be deployed and run in containers. The necessary files and configurations for creating the Docker container that hosts DoctorAI on Google Cloud Run are included in this component. These files ensure that the application is properly packaged and ready to be deployed on the cloud infrastructure.
6. **Telegram Bot for User Interface** The Telegram Bot serves as the front-end for DoctorAI, enabling users to interact with the system using natural language. Users can send messages to the bot, ask medical questions, and receive diagnoses and recommendations in response. The Telegram Bot handles the communication between users and the backend system.

With these key components, DoctorAI can provide medical diagnoses and recommendations to users 24/7, leveraging quality medical knowledge from NHS.UK, and ensuring the information provided is accurate and trustworthy, thanks to the creator's review and control of the data. The deployment on Google Cloud Run enables scalability and availability, allowing the bot to handle user requests efficiently.

* 1. **Medical Knowledge Source**

Here is the URL we download the data from: <https://www.nhs.uk/conditions/>

* 1. **Data Scraping Program**

The program name is: NHS\_Content\_Extraction\_Final.py

Please ensure correct drive will be setup as:

|  |
| --- |
| documents\_directory = '/nhs\_source'  input\_folder = documents\_directory  output\_folder = "/nhs\_content" |

The ‘/nhs\_content folder will be needed to collect the sub web page names needed for second downloading.

The final result will be downloaded into /nhs\_content folder.

A copy of final downloading content will be provided in the **nhs\_content.zip**

* 1. **Embedding and Vector DB Creation Program**

The program name is: doctorai\_embedding\_final.py

Please ensure correct drive will be setup as:

|  |
| --- |
| path\_content\_source = "/nhs\_content"  path\_vector\_db = "/nhs\_vector\_db" |

Please also ensure that OPENAI Credential for the embedding and QnA test:

|  |
| --- |
| os.environ['OPENAI\_API\_KEY'] = "NA" |

After you running the program, the four vector DB will be created in path\_vector\_db folder. A copy of that vector db were included in the zip file: **nhs\_vector\_db.zip**

* 1. **Main Telegram Bot and QnA Program**

The program name is: doctorai\_telegram\_final.py

The file will be combined with other file to be deployed o GCP. Ensure the folder was :

|  |
| --- |
| path\_vector\_db = "/app” |

* 1. **Files Used to Create the Docker for Google Cloud Run**

Following files will be used to build the docker, and will be used for Google Cloud deployment.

Here the zip file name for docker submission: **Docker Submission\_DoctorAI.zip**

Just be aware that if you want to deploy the docker file to GCP environment, additional project id will be needed. Please check with google reference below for Cloud Run deployment: <https://cloud.google.com/run>

* 1. **Telegram Bot for User Interface**

User can use telegram bot father to create the chat bot. Please ensure the bot API credentials will be captured for future usage in the program 1.4.

Apart from it, please use Postman API to ensure web hook setup to get telegram push the message to the API point created in 1.5, after cloud run environment being ready.

Here is details of web hook API point setup url: https://core.telegram.org/bots/api#setwebhook

A screenshot of a computer

Description automatically generated

That’s all for DoctorAI.

1. **Ultrasound Education Assistant**

The structure of Ultrasound Education Assistant is very similar to Doctor AI. The only difference is the data extraction was avoided as the source data of ultrasound training material was provided by Butterfly Network.

* 1. Source Data

The source data fine name is: **education.csv**

* 1. Embedding and Vector DB creation

Here are some folders will be used for source file and vector DB. Please ensure copy the education.csv in the source folder:

|  |
| --- |
| path\_content\_source = "/nhs\_content\_2"  path\_vector\_db = "/nhs\_vector\_db4" |

The file name : **ultrasound\_edu\_embedding\_final.py**

* 1. Main Chatbot Telegram Program and Docker File

Filling files will be needed in the docker file for the deployment:

A screenshot of a computer

Description automatically generated

The main.py is the program handle the telegram and qna function. Please ensure OPENAI and Telegram credentials was included in the code. The Zip file of docker was attached:

**Docker Submission\_EDU.zip**

* 1. **Telegram Bot for User Interface**

User can use telegram bot father to create the chat bot. Please ensure the bot API credentials will be captured for future usage in the program 2.3.

Apart from it, please use Postman API to ensure web hook setup to get telegram push the message to the API point created in 2.3 , after cloud run environment being ready.

Here is details of web hook API point setup url: https://core.telegram.org/bots/api#setwebhook

A screenshot of a computer

Description automatically generated

That’s all for Ultrasound Education Assistant.

Here is my email address for further questions: jeff\_z\_xu@yahoo.com

Good Luck !