1. Please describe the architecture of the current chatbot system. Identify the components and check where are they running now.

1.1 User Interface (UI):

Our chatbot UI is using Telegram chat as a client, the screenshot as below.

A screenshot of a phone

Description automatically generated

1.2 Input Processing:

typing

1.3 Chatbot Engine / Core:

import/pip

1.4 Response Generation

Firstly, it processes the input into a format than the model can read, then tokenization the pre-processing data. Secondly, the model will try to understand that content which based on the transformer. thirdly, it will generate response via strategies decode and processing the post. Finally, the output will return to the client-end which make it readable.

1.5 Integration Layer

In the understand of context, they used embedding function to vectorize the content and calculating Cos similarity.

1.6 Backend Services

My chatbot based on Pass, which use cloud service from OpenAI API, Our university had brought their service to construct a educational ChatGPT, moreover they provide HKBU ChatGPT's API to let our student access it.

1.7 Training and Evaluation

Currently, our Lab3 and 4, did not require to Training and evaluation the GPT model which we implemented.

2. Explain how do your chatbot handle the special command. You need to trace the code and explain that.

In chatbot.py. we customized the special Q&A by following command:

dispatcher.add\_handler(CommandHandler("hello", kevin\_command))

def kevin\_command(update: Update, context: CallbackContext) -> None:

"""Send a message when the command /hello Kevin"""

update.message.reply\_text('Good day, Kevin!')

Using dispatcher function can implement this special function.

3. Update your code so that when user type /hello Kevin , it will reply Good day, Kevin! . Write down the change you have made.

dispatcher.add\_handler(CommandHandler("hello", kevin\_command))#Corresponding to below.

def kevin\_command(update: Update, context: CallbackContext) -> None:

"""Send a message when the command /hello Kevin"""

update.message.reply\_text('Good day, Kevin!')



4. Make a few screen caps to prove that you have applied your own Redis account, used it in your chatbot, and push the code on GitHub (at least 2 commits - lab3/lab4).

A screenshot of a computer

Description automatically generated