Documentation: LeuphAI Q&A

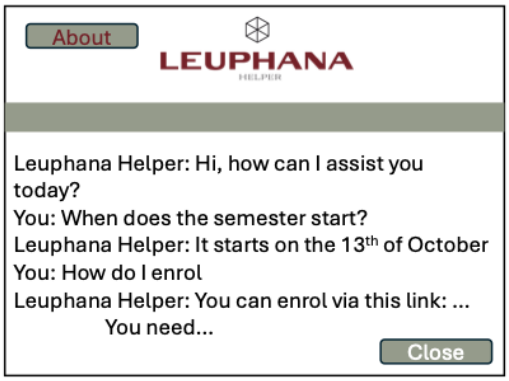
tech basics final project Lennox Krause

Initial idea

- A useful AI assistant for students studying at Leuphana
- Using a preexisting AI model and implementing it into python
- Finetuning its answers and training it
    - Using Leuphana website?
    - Found website crawler in internet, looks quite complicated, but possible?
- Maybe implementing it into some kind of interface if it's not enough coding by then

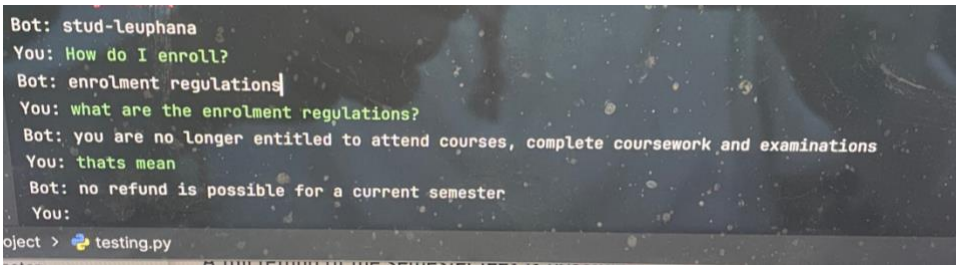
Initial code structure for idea

- Typical While loop
- Player inputs and bot answers
- Possibility to exit
- Code for finding the right context
- Website crawler
- Context file?
- Pygame or TKinter interface for better graphics and more coding?
- Fine tuning stuff (I still need to learn details)



Initial Research:

- Asked ChatGPT and created a small demo version with its help



- Got more resources for learning AI implementation in python from both ChatGPT and our seminary teacher
- Wanted to find a data security conform AI system
    - Found EuroLLM, but got the feedback that it would not work very good for this task

Development process

- How can I improve the output of my AI?
    - Better context data
    - Algorithms for processing context data
    - Better suited AI model
- First idea: improving the output with better context data -> WebCrawler and also some processing attempts
    - Found a tutorial on creating a WebCrawler on the OpenAI website (1)

        - After many hours, the WebCrawler I had created pulled a lot of text from the website, but the processing did not work out quite right
        - I tried out a few different ways to process the data, from the OpenAI website and elsewhere, but either the context was too processed with no data left, or too big with too much data for my laptop to handle
        - So, after wasting a lot of time, I chose to copy the data myself, and to only focus on one aspect of the website (Leuphana music centre)
        - After a bit of trial and error, I restructured the context from the website manually to make it even more readable
    - Second idea: better suited AI model
        - First, I exchanged the EuroLLM model with Roberta, a model from Germany
        - Then, I got the idea to mix Q&A answer finding with text generation to create better suited texts
        - This may have increased the time it takes until the answer is generated, but it also immensely improved the quality of the answers
    - Creating a UI
        - I also wanted to implement some TKinter to do some more actual coding, as the goal of this final project task is also to showcase the knowledge on python I gained attending this course
        - Mainly used one website explaining almost everything needed to understand TKinter (2)

Future improvements and known limitations

- Biggest known limitation: only focusing on Leuphana music centre
    - Two solutions: Either increasing the context and using a device with more computing power or implementing some kind of live WebCrawler which finds context for the answer while its being asked
- Implementing more UI options
    - A “help”/about button giving background information about the chatbot
    - Different tabs for different conversations
- HTML embedding for using it on the Leuphana website

Most important references

(1) https://platform.openai.com/docs/tutorials/web-qa-embeddings
(2) https://www.pythontutorial.net/tkinter/tkinter-hello-world/
https://huggingface.co/learn/llm-course/chapter1/1