Investigating the Perceived Effectiveness of ChatGPT in Facilitating Learning

Hon Keung YAU
City University of Hong Kong. Department of Systems Engineering. Kowloon Tong, Kowloon, Hong Kong
honkyau@cityu.edu.hk

Ka Fai TUNG
City University of Hong Kong. Department of Systems Engineering. Kowloon Tong, Kowloon, Hong Kong
kftung4-c@my.cityu.edu.hk

ABSTRACT
This study explores the development and evaluation of a chatbot model designed to facilitate learning within a department of a university. The project aims to enhance the learning experience by incorporating customized data into the chatbot's knowledge base, enabling personalized and context-aware interactions. The research investigates the effectiveness of the chatbot model in improving learning outcomes, enhancing user engagement, and providing tailored support to students in the department.
The development of the chatbot model is detailed, including technology selection, data acquisition, preprocessing, and model architecture. The study also demonstrates creating web applications by Streamlit, Vector Database by Supabase, and graphical user interface by Tkinter.
The chatbot model demonstrates its effectiveness in enhancing the learning experience within the department. By continuously refining the model and exploring advancements in technology, chatbots have the potential to become integral tools in future educational practices, providing personalized and engaging learning experiences.
Keywords: ChatGPT, Chatbot model, Steamlit

INTRODUCTION
The development of chatbot models has been greatly influenced by existing systems, such as the chatbot provided by the City University of Hong Kong. These systems have demonstrated the potential of using chatbots to enhance user experiences and provide valuable services. The City University chatbot, powered by the Microsoft Azure OpenAI (City University of Hong Kong, n.d.), has shown the effectiveness of utilizing advanced language models to engage in natural language conversations with users.

However, while the City University chatbot relies solely on the ChatGPT (City University of Hong Kong, n.d.), there is an opportunity to create a chatbot that goes beyond generic responses and incorporates customized data. By leveraging customized data, the chatbot can provide users more targeted and tailored information, enhancing the overall user experience.

The motivation behind creating a chatbot with customized data is the need for a more personalized and context-aware conversational agent. Traditional chatbots often struggle to understand specific domain knowledge or provide specialized information. By incorporating customized data, the chatbot can address this limitation and offer more accurate and relevant responses to user queries.

The project aims to develop a chatbot that utilizes a combination of advanced language models and domain-specific data. Integrating customized data into the chatbot's knowledge base can provide specialized information and cater to specific user needs. This approach allows for a more comprehensive and accurate understanding of user queries, leading to more meaningful and contextually relevant responses.

Customized data also opens up opportunities for the chatbot to engage in domain-specific discussions and provide targeted recommendations. For example, in an educational context, the chatbot can leverage course materials, textbooks, or research papers to offer in-depth explanations, suggest relevant resources, and guide learners through specific topics. This personalized approach enhances the learning experience and provides users with valuable, tailored information.

Industry developments and research studies have highlighted the value of incorporating customized data into chatbot models. By leveraging domain-specific knowledge and data, chatbots can provide more accurate information, enhance user engagement, and improve overall performance. The project aims to build upon these advancements and contribute to the growing field of chatbot development.

The project background emphasizes the influence of existing chatbot systems. However, the project seeks to go beyond generic responses by incorporating customized data into the chatbot model. This approach aims to provide more personalized and context-aware interactions, catering to specific user needs and offering specialized information. By leveraging customized data, the project aims to enhance the user experience, improve response accuracy, and contribute to the advancement of chatbot technologies.

METHODOLOGY

This section outlines the methodology employed in developing the chatbot model for facilitating learning. The steps involved in the development process are described below:

1. Interacting with the OpenAI API

The research team leveraged the capabilities of the OpenAI API to develop the chatbot model. The API allowed the team to interact with a powerful language model and utilize its natural language processing capabilities. By sending prompts and receiving model-generated responses, the team could engage in a dialogue with the chatbot and incorporate its responses into the model's training data.

2. Collecting and preparing data from the department website

Data was collected from the department website to train the chatbot model. This data included frequently asked questions, course information, and other relevant resources. The collected data underwent a preprocessing stage to ensure its suitability for training the chatbot model. Preprocessing techniques such as removing HTML tags, normalizing text, and handling duplicates were applied to enhance the quality and consistency of the data. This preparation step aimed to improve the chatbot's understanding and generation of appropriate responses.

3. Creating the bot

The development of the chatbot involved designing the bot's architecture, algorithms, and underlying technologies. The architecture was designed to enable efficient information retrieval and response generation. The team implemented algorithms for natural language understanding and generation, which allowed the chatbot to comprehend user queries and generate relevant and coherent responses. Design decisions were made to balance the chatbot's responsiveness, accuracy, and user-friendliness. These decisions were guided by the research objectives of facilitating learning and providing personalized support to users.

4. Creating the user interface

The user interface for the chatbot was developed to ensure a seamless and intuitive user experience. The interface design incorporated user-centered principles, aiming for simplicity, clarity, and ease of use. Considerations were made to enhance usability, such as providing clear instructions, intuitive navigation, and appropriate visual cues. The user interface implementation focused on integrating the chatbot model's functionalities into a user-friendly interface, enabling users to interact with the chatbot easily and access relevant learning resources.

In addition to the specific steps mentioned above, the overall methodology of the research study employed various techniques for data collection, evaluation, and assessment of the chatbot model's effectiveness. Data collection involved gathering user interactions and feedback, which provided valuable insights into the users' experiences and perceptions of the chatbot. Evaluation techniques included pre- and post-testing to assess learning outcomes, user surveys to measure user satisfaction and engagement, and usage analytics to track patterns of interaction.

The chosen methodology aligns with the research objectives of evaluating the effectiveness of the chatbot model in facilitating learning. By collecting user data, the study aims to gain a comprehensive understanding of the chatbot's impact on learning outcomes and user engagement. The evaluation techniques employed provide quantitative and qualitative measures to assess the effectiveness of the chatbot model in achieving its intended purpose.

Overall, the methodology employed in developing the chatbot model includes interacting with the OpenAI API, collecting and preparing data from the department website, creating the chatbot and its user interface, and employing various data collection and evaluation techniques. This methodology ensures a systematic approach to the development and assessment of the chatbot model, facilitating the achievement of the research objectives.

DEVELOPMENT

1. *Interacting with the OpenAI API*

The first step of the dissertation's model development session focused on interacting with the OpenAI API, specifically utilizing the GPT-3.5 model. In order to explore the capabilities of the API, a website application was developed using Streamlit (Streamlit Inc., 2024). This application serves as a chatbot powered by the GPT-3.5 model, allowing users to engage in natural language conversations.

By integrating the OpenAI API into the website application (OpenAI API, n.d.), the performance of the GPT-3.5

model was tested and evaluated in a practical setting. This involved sending prompts or queries from the users to the API and receiving responses generated by the model. The interactions between the users and the chatbot provided valuable insights into the model's language understanding, response generation, and overall conversational abilities.

Furthermore, the fine-tuning feature offered by the OpenAI API was investigated (OpenAI API, n.d.). To employ this functionality, relevant data from the department website was gathered, ensuring it was representative of the domain or topic of interest. This data collection process involved organizing and curating the information to create a suitable training dataset.

Once the dataset was prepared, it was uploaded to the OpenAI API to initiate the fine-tuning process. This involved training a specialized model that was specifically tailored to better understand and respond to queries related to the department. By fine-tuning the base GPT-3.5 model with domain-specific data, the objective was to enhance the chatbot's performance within the context of the department.

Overall, this initial phase of the dissertation's model development session was to evaluate the OpenAI API's capabilities by building a chatbot application and testing it with the GPT-3.5 model. Additionally, the potential of fine-tuning the model using domain-specific data was explored, specifically collected from the department website, in order to improve the chatbot's performance within the targeted domain.

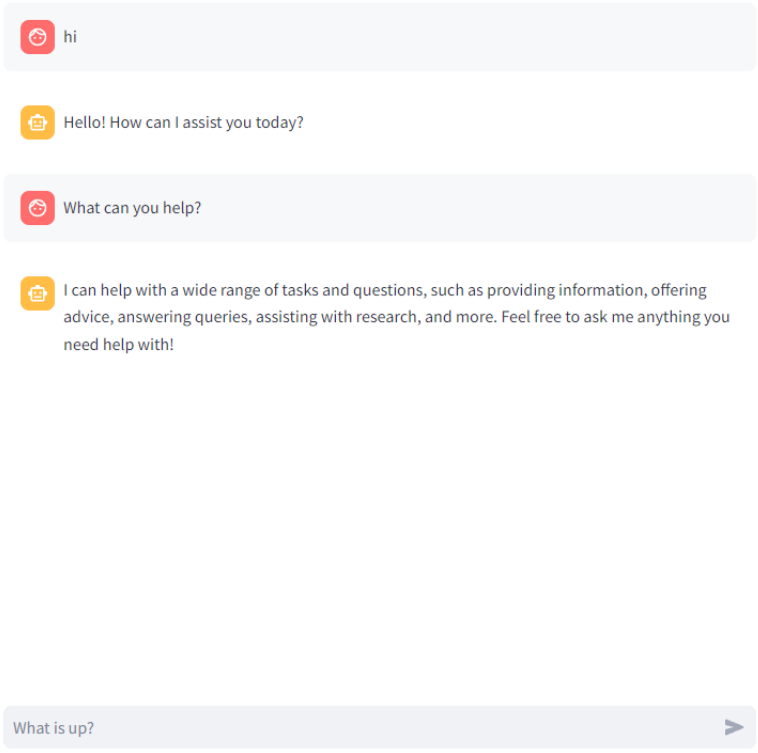

Figure 1 Website application developed using Streamlit

2. *Collecting and preparing data from the department website*

The second step of my dissertation's model development session focused on collecting and preparing data from the website at https://www.cityu.edu.hk//. In order to gather the necessary information, web scraping techniques were employed using Python Selenium (Muthukadan, 2024).

However, during the web scraping process, challenges were encountered due to the irregular format of the department's website. This irregularity made it difficult to apply a uniform scraping approach across all sections of the website. As a result, a combination of web scraping and manual work was resorted to ensure comprehensive data collection.

The specific sections on the website that were of interest to my research were identified: People, Programmes, Student Life, Laboratories, Job Opportunities, and Contact Us. These sections contained valuable information

related to the department's faculty, academic programs, student activities, research laboratories, employment opportunities, and contact details.

Scraping scripts tailored to each section were implemented using Python Selenium, extracting relevant text and data. However, due to the irregular structure of the website, manual intervention was necessary to handle certain sections where automated scraping proved challenging.

A text file was chosen to use to organize and store the collected information, which served as a convenient and flexible format for data storage. The extracted information from each section, ensuring accuracy and completeness, was meticulously recorded.

By successfully collecting data from the identified sections of the website, valuable insights and content were obtained that could be utilized for various purposes within the dissertation's model development. This data would serve as the foundation for training and fine-tuning the GPT-3.5 model, enabling it to provide accurate and relevant responses within the domain of the department.

Overall, the data collection process involved a combination of web scraping using Python Selenium and manual work to overcome the irregular format of the department's website. The collected information was stored in a notepad, providing a centralized repository of data for further processing and analysis in the dissertation.

3. Creating the bot

The third step of my dissertation focused on developing a conversational bot capable of retrieving relevant information based on user input. To accomplish this, the OpenAI API was utilized to embed text and Supabase as a Vector Database.

In the previous step, a substantial amount of data that would serve as the basis for the bot's knowledge was collected. This information needed to be transformed into a suitable format for comparison and retrieval. Moreover, the OpenAI API's embedding capability was employed, specifically utilizing the "text-embedding-ada-002" model. This model converted textual information into vector representations, which capture the semantic meaning of the text.

Once the data was transformed into vector form, both the original information and its corresponding vector representation in a Vector Database provided by Supabase (Supabase Inc., n.d.) were stored. It offers a convenient platform for managing and querying vector-based data efficiently.

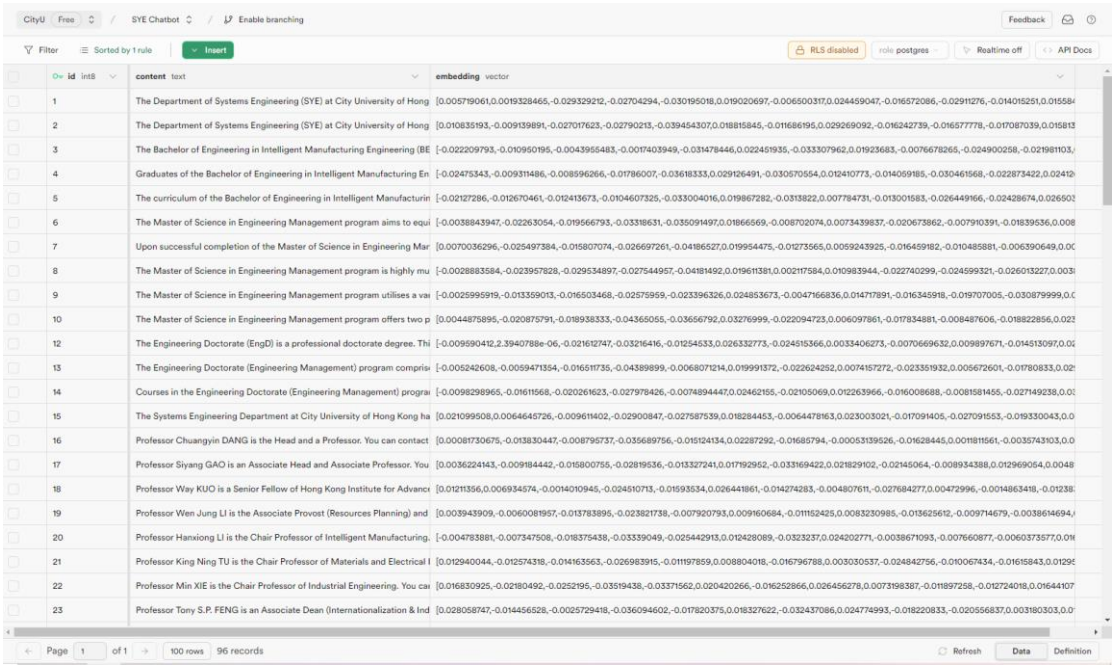

Figure 2 Vector Database created using Supabase

A process that begins with user input was designed to retrieve relevant information from the database. When a user

interacted with the bot, their input was first sent to the OpenAI API to perform text embedding. The API converted the input into a vector representation using the same "text-embedding-ada-002" model.

After obtaining the vector representation of the user's input, a pre-defined function in Supabase was invoked to compare this vector with the vectors stored in the database. The function identified the most closely related information entries by measuring the similarity between vectors.

The output of this process was then provided to the user as the bot's response, comprising the most relevant information retrieved from the database. However, if the similarities between the user's input vector and the stored vectors fall below a pre-set threshold, it indicates a lack of closely related information. The user was notified that no relevant information could be found in such cases.

This approach allowed the bot to effectively retrieve and present information based on user queries, leveraging the power of text embedding and vector comparison. By combining the capabilities of the OpenAI API and Supabase, a robust system has been developed to create an intelligent conversational bot that could provide meaningful responses to user inquiries.

4. Creating the graphical user interface

The fourth step of the dissertation focused on developing a graphical user interface (GUI) to enhance the usability and accessibility of the conversational bot. This step involved utilizing Tkinter (Python Software Foundation, 2024), a Python library, to design and implement the GUI application.

To begin, icons and logos were incorporated into the GUI application to create a visually appealing and recognizable interface. These visual elements help users identify and connect with the bot, giving the application a more professional and polished appearance, as shown in Figure 3.

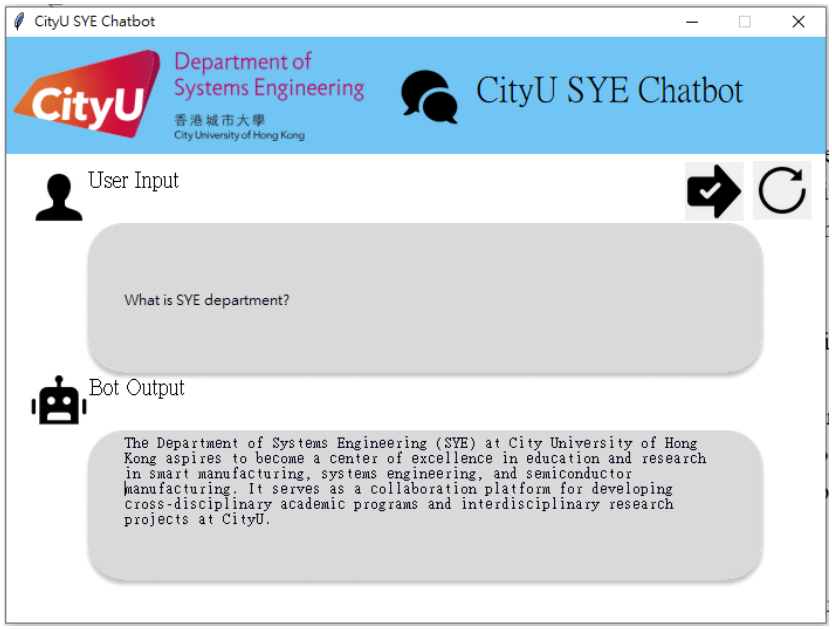

Figure 3 GUI application created using Tkinter

Within the GUI, users can input their questions or queries into a designated text input box. This input serves as the user's interaction with the bot and is the starting point for retrieving relevant information. The text input box allows users to enter their queries in a user-friendly manner easily.

The functions developed in the previous step were integrated into the GUI application to facilitate the retrieval of information. These functions are triggered when the user presses the submit button after entering their query. The functions take the user's input, convert it into vector form using the OpenAI API, and compare it with the vectors stored in the Supabase Vector Database. The result of this comparison is then displayed in another text box within the GUI.

The Python program was converted into an executable file (.exe) using the "auto-to-py-exe" module (Vollebregt,

2024) to enhance the usability and distribution of the application. This conversion allows users to run the application without having to install Python on their computers, making it more accessible to a wider audience.

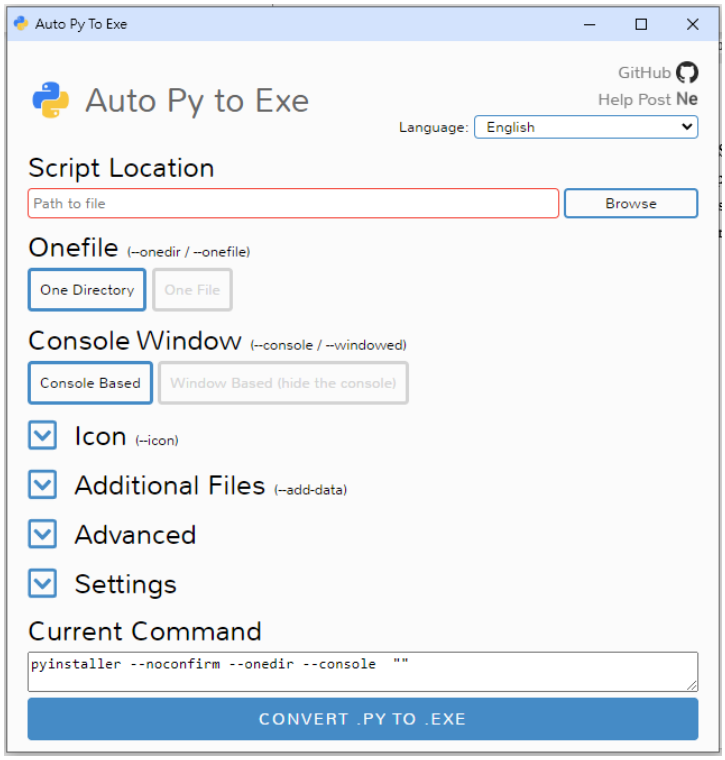

Figure 4 Python auto-to-py-exe module

In addition, an installer was created for the entire application using NSIS (Nullsoft Scriptable Install System) (NSIS, 2024). This installer ensures that all the necessary components and files are packaged together so users can install the application seamlessly without missing any dependencies. The installer simplifies the installation process and provides a smooth user experience.

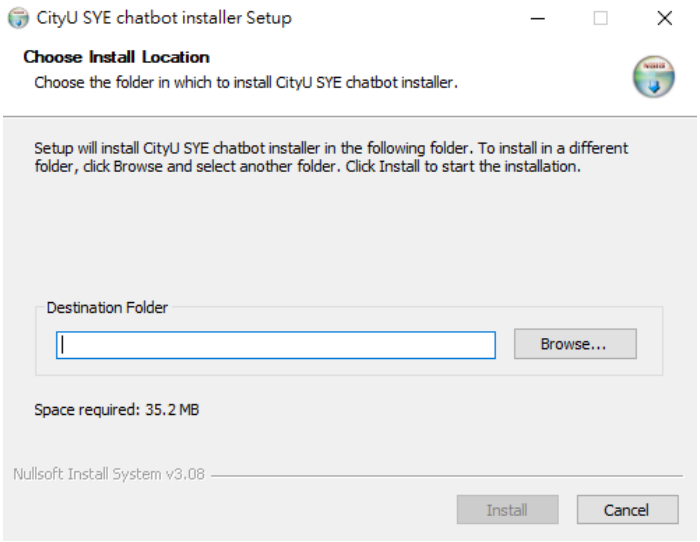

Figure 5 Chatbot installer created using NSIS

By creating a user-friendly GUI, converting the program to an executable file, and generating an installer, the conversational bot application has been made more accessible, convenient, and user-friendly. These steps ensure that users can interact with the bot effortlessly without any technical barriers or installation concerns.

ANALYSIS

The analysis section of this dissertation report focuses on evaluating the effectiveness of the chatbot model developed for facilitating learning. The key findings and insights obtained from the evaluation process are highlighted, emphasizing the impact and implications of the chatbot model on the learning outcomes of the target audience.

One of the primary objectives of this research was to explore the capabilities of the OpenAI API, specifically the GPT-3.5 model, in developing a chatbot for facilitating learning. Through the integration of the API into a website application, users were able to engage in natural language conversations with the chatbot. The evaluation of the chatbot's performance provided valuable insights into the model's language understanding, response generation, and conversational abilities.

The findings from the evaluation process indicated that the GPT-3.5 model demonstrated strong language understanding capabilities. It was able to comprehend and interpret user queries accurately by embedding user queries into vector form, allowing for meaningful and contextually relevant responses. The chatbot then provides users with the information they were seeking.

Moreover, the evaluation revealed that the chatbot's conversational abilities were effective in facilitating learning. Users were able to engage in interactive conversations with the chatbot, asking questions and receiving informative responses. The chatbot acted as a virtual assistant, getting users the most relevant resources and information from the Vector Database.

The evaluation process also highlighted the impact of fine-tuning the GPT-3.5 model using domain-specific data. By training the model with data collected from the website, the chatbot's performance within the context of the department was significantly improved. The fine-tuned model demonstrated a deeper understanding of -related topics and was able to provide more accurate and specialized responses.

Overall, the analysis of the chatbot model's effectiveness in facilitating learning revealed several key findings. The GPT-3.5 model exhibited strong language understanding capabilities and was able to generate coherent and contextually relevant responses. The chatbot's conversational abilities effectively facilitated learning, providing personalized recommendations and resources to users. Additionally, the fine-tuning of the model using domain-specific data enhanced its performance within the targeted domain.

The implications of these findings are significant in the field of education and learning. The development and implementation of chatbot models for learning purposes can offer a scalable and accessible solution to support learners in acquiring knowledge and navigating educational resources. The personalized nature of the chatbot's interactions enhances user engagement and promotes effective learning outcomes.

In conclusion, the evaluation of the chatbot model developed for facilitating learning demonstrated its effectiveness in supporting learners and providing valuable educational resources. The strong language understanding capabilities, personalized recommendations, and the impact of fine-tuning the model within a specific domain contribute to its potential as a valuable tool in the field of education. Further research and development in this area can lead to advancements in intelligent learning systems and enhance the overall learning experience for students.

CONCLUSION

In conclusion, the evaluation of the chatbot model developed for facilitating learning has provided valuable insights into its effectiveness and impact on the learning outcomes of students. The analysis revealed that the chatbot demonstrated strong language understanding and response generation capabilities, offering accurate and contextually appropriate answers to user queries. Its integration into the learning process had several positive implications, including instant access to relevant information, the promotion of self-directed learning, and the creation of an interactive learning environment. However, there is still room for improvement in the chatbot's performance, particularly in understanding complex queries and providing nuanced responses. Overall, the findings highlight the potential of chatbot technology in enhancing educational experiences and suggest avenues for further research and improvement.

The chatbot model proved to be a valuable resource for students, providing instant access to relevant information and saving time and effort in searching for resources. By leveraging the vast knowledge base of the website, the chatbot efficiently retrieved specific information on faculty, programs, student life, laboratories, job opportunities, and more. This instant access enhanced the learning experience, allowing students to quickly find answers to their questions and explore various aspects of the department. The chatbot can provide personalized guidance and

support promoted self-directed learning, empowering students to engage in the learning process and clarify doubts actively. The conversational nature of the chatbot facilitated a more interactive and dynamic learning environment, fostering deeper understanding and exploration of topics of interest.

While the chatbot model demonstrated promising performance, there are areas that require further improvement. In some instances, the chatbot struggled to understand unrelated queries or provide nuanced responses. To address these limitations, fine-tuning the model with additional data and incorporating user feedback is recommended. Fine-tuning the model using a larger dataset of domain-specific information from the department could enhance its domain knowledge and improve its response accuracy. Furthermore, actively seeking user feedback and iteratively refining the model based on user interactions can enhance its conversational abilities and adaptability to user needs. Regular updates and maintenance of the chatbot's Vector Database are essential to ensure that it remains up-to-date with the department's evolving information and requirements.

In addition to improving the chatbot model, future research can explore several avenues to further enhance the effectiveness of chatbots in facilitating learning. Firstly, integrating natural language processing techniques and machine learning algorithms can enable the chatbot to understand and generate more complex responses. This could involve leveraging pre-trained models, such as GPT-3.5, and exploring advancements in language models to enhance the chatbot's capabilities. Secondly, incorporating multimodal capabilities, such as text-to-speech and speech recognition, can enable the chatbot to cater to different learning preferences and accessibility needs. By providing audio-based responses or accepting voice inputs, the chatbot can offer a more inclusive learning experience. Additionally, integrating the chatbot into learning management systems or educational platforms can further enhance its accessibility and seamless integration into existing educational workflows.

Furthermore, the impact of the chatbot model on learning outcomes can be assessed through longitudinal studies and comparisons with traditional learning approaches. Conducting follow-up studies to evaluate the long-term effects of the chatbot on students' knowledge retention, academic performance, and overall learning experience can provide deeper insights into its effectiveness. Comparing the performance and satisfaction of students using the chatbot with those relying solely on traditional learning resources can help assess the added value and benefits of the chatbot in educational settings.

The evaluation of the chatbot model developed for facilitating learning has demonstrated its effectiveness in providing timely and relevant information, promoting self-directed learning, and creating an interactive learning environment. The findings underscore the potential of chatbot technology in enhancing educational experiences and improving learning outcomes. By fine-tuning the model, incorporating user feedback, and exploring new avenues for improvement, chatbots can continue to evolve as valuable tools for facilitating learning in various educational contexts. As technology advances and research progresses, chatbots have the potential to become integral components of future educational practices, supporting learners in their quest for knowledge and fostering a more engaging and personalized learning experience.

REFERENCES

City University of Hong Kong. (n.d.). *CityU GPT Chatbot Service*. Retrieved June 3, 2024, https://www.cityu.edu.hk/GenAI/#cityu-gpt

Muthukadan, B. (2024). *Selenium with python*¶. Selenium with Python - Selenium Python Bindings 2 documentation. Retrieved June 3, 2024, from https://selenium-python.readthedocs.io/

NSIS. (2024, March 30). *Main page*. NSIS Wiki. Retrieved June 3, 2024, from https://nsis.sourceforge.io/Main_Page

OpenAI API (n.d.). Introduction. Documentation. Retrieved June 3, 2024, from https://platform.openai.com/docs/introduction

Python Software Foundation. (2024, June 2). *Graphical user interfaces with TK*. Python documentation. Retrieved June 3, 2024, from https://docs.python.org/3/library/tk.html

Streamlit Inc. (2024). Build a basic LLM chat app. Documentation. Retrieved June 3, 2024, from https://docs.streamlit.io/develop/tutorials/llms/build-conversational-apps

Supabase Inc. (n.d.). *Main menu*. Supabase Python Client - Introduction. Retrieved June 3, 2024, from https://supabase.com/docs/reference/python/introduction

Vollebregt, B. (2024, March 25). *Auto-py-to-exe*. PyPI. Retrieved June 3, 2024, from https://pypi.org/project/auto-py-to-exe/