UCSI University®

FACULTY OF ENGINEERING, TECHNOLOGY & BUILT ENVIRONMENT

BER2033: INTRODUCTION TO ARTIFICIAL INTELLIGENCE

REPORT (10%)

PROJECT TITLE : Local AI Chatbot

PREPARED BY : Mohammed Tariq Mohammedelhassan (1002371596)

Ryan Jong Ntambwe (1002161713)

Ryan H Chiminya (1002370502)

SEMESTER : Jan – Apr 2025

INSTRUCTOR : Dr. Mastaneh Mokayef

Learning Outcome (A)	Poor (1 point)	Fair (2 points)	Moderate (3 points)	Good (4 points)	Excellent (5 points)	Point (B)	Total (B/5 x A%)
(a) Introduction (10%) **PLO2: Problem Analysis Knowledge Profile (WK): WK 1**	Introduction is missing or lacks clarity and relevance.	Introduction provides basic background information but lacks a clear objective or purpose.	Introduction provides a clear objective and purpose but may lack some background information.	Introduction provides a clear objective, purpose, and relevant background information.	Introduction is engaging, clearly states the objective, purpose, and provides comprehensive background information.		
(b) Methodology (30%) **PLO2: Problem Analysis Knowledge Profile (WK): WK 1**	The model training process is ineffective, leading to poor model performance. Minimal or no fine-tuning.	Some effectiveness in the model training process, however improvements are needed. Some fine-tuning efforts made.	The model training process is moderately effective, resulting in satisfactory performance. Moderate fine-tuning efforts made.	The model training process is effective, resulting in good model performance. Significant fine-tuning efforts made.	The model training process is exceptionally effective, resulting in excellent model performance. Extensive fine-tuning efforts made.		
(c) Results and Discussion (30%) **PLO2: Problem Analysis Knowledge Profile (WK): WK 1**	Model accuracy is consistently low. Confusion matrices are incomplete or unclear.	Model accuracy is somewhat inconsistent. Confusion matrices are somewhat complete and clear.	Model accuracy is adequate with minor inconsistencies. Confusion matrices are adequately complete and clear.	Model accuracy is consistent and satisfactory. Confusion matrices are clear and complete.	Model accuracy is consistently high. Confusion matrices are highly clear and provide valuable comprehensions.		

Learning Outcome	Poor (1 point)	Fair (2 points)	Moderate (3 points)	Good (4 points)	Excellent (5 points)	Point (B)	Total (B/5 x A%)
(d) Conclusion (10%) **PLO2: Problem Analysis Knowledge Profile (WK): WK 1**	Conclusion is missing or lacks a clear summary.	Conclusion provides a summary but lacks reflection or future considerations.	Conclusion summarizes the main findings but lacks deep reflection or future considerations.	Conclusion provides a concise summary and reflects on the findings and implications.	Conclusion is comprehensive, provides a clear summary, and offers insightful reflections and future considerations.		
(e) Recorded Demo Video (20%) **PLO2: Problem Analysis Knowledge Profile (WK): WK 1**	Demonstration is unclear or incomplete.	Demonstration is somewhat clear and complete.	Demonstration is clear and complete.	Demonstration is highly clear and complete.	Demonstration is exceptionally clear and complete.		
Total Marks (10%)							

Table of Content

Table of Contents

Introduction

Artificial intelligence-driven chatbots have greatly changed the system of human-machine interaction from simple scripted conversations to highly advanced ones based on natural language processing. Chatbots are now ubiquitous as essential parts of various industries, with applications in customer services, personal assistance, as well as generating content, as they simulate human interaction to enhance the overall experience of users.

Although more functional, most of these chatbot systems are reliant on internet connectivity, with this placing them under restriction where used offline. The dependency on these cloud-based infrastructures creates challenges in areas with limited network accessibility or where high data privacy needs are to be met. Consequently, there's a need for chatbots that can fully run offline, yet still provide accurate and contextually appropriate responses.

This paper presents a design of a localized chatbot based on offline connectivity, using Ollama AI library. The aim of this design primarily lies in creating a chatbot designed to provide quick, secure, and offline communication, thus spreading its usability to remote areas, local networks, or where data sensitivity 123takes precedence. The choice of the code base as Python was based on its ease, versatility, and strong backing of artificial intelligence growth, thus well suited to its deployment in offline artificial intelligence programs.

The chatbot uses the Ollama AI library, whose offline AI functionalities are brought to users by contrast with many other libraries with ongoing internet connection requirements. With this feature, Ollama becomes highly adaptable to autonomous system development due to its offline functionalities. The chatbot mainly uses the Mistral model, which has high-speed and efficient replies; however, LLama3 and Gemma are also supported as alternative models. LLama3 has better language understanding abilities, while Gemma allows more advanced conversational interaction with a more human-like interaction experience. With this kind of support, the chatbot becomes adaptable to various conversational needs based on user preference.

In short, this local chatbot program intends to create a robust offline conversation system. Through leveraging the Ollama AI library coupled with Python's advanced toolset, it shows that AI programs are viable with regard to being internet-access independent, seeking to solve the increasing demand for independent and personal systems.

Scope of the Project

The scope of this project includes conceptualization and development of a local chatbot with offline functionality, thus independent of internet connectivity. The ultimate aim is to make a chatbot available with immediate, secure, and contextually correct responses in an offline setting. The approach caters to certain limitations inherent in traditional cloud-based chatbots, which often fail in situations where offline performance or data privacy considerations are mandates.

The chatbot has been designed for deployment in a diversity of situations with limited, unreliable, or adverse Internet connectivity. They are situations such as remote rural areas, secure institutions with data privacy requirements, and local networks whose design will seek to limit external exposure. Also, in academic situations, there are opportunities with the chatbot as an interactive study tool unproblematic with respect to web services.

The major components of the project are harnessing Python as the choice of programming language due to its robust reliability and huge library base for artificial intelligence operations. The Ollama AI library was chosen based on its ability to run without an internet connection, while Mistral was used due to its quick response nature. The LLama3 and Gemma are supported by the chatbot to provide users with advanced or human-like interaction possibilities. Future possibilities include adding a graphical user interface (GUI), voice input, and external API compatibility to enhance usability as well as overall performance.

Methodology

Simple Overview

1. Technology Stack:
    - Programming Language: Python
    - AI Library: Ollama
    - AI Model: Mistral (Primarily), LLama3, Gemma
2. Working Mechanism:
    - The chatbot prompts the user to enter a message.
    - The user input is stored in a conversation history list.
    - The chatbot processes the input using the specified AI model via the Ollama library.
    - The response is generated based on the conversation history, maintaining continuity in responses.
    - The output is displayed to the user.
3. Code Implementation:
    - The chatbot runs locally, without internet access, using the Ollama library for AI-driven responses.
    - The chatbot continuously accepts user input until the keyword 'quit' is entered.

Detailed Walkthrough

The methodology used in building local chatbot followed a systematic iterative methodology with a focus on building an offline conversational agent using Python together with Ollama AI library. The major aim was to build a chatbot with offline functionality independent of connectivity with the World Wide Web yet still provide well-coherent and contextually equivalent responses. The iterative approach allowed reaching a product capable of meeting offline requirements, as well as performance requirements and interactive usability.

The first phase of the methodology was focused on a close study of the requirements of the project. The phase was designed specifically to understand the challenges of building a chatbot independent of internet connectivity, as compared to other present chatbots based on cloud architecture. The offline-capability requirement was identified as the critical criterion, paving the way for seeking an artificial intelligence library with offline connectivity features.

Ollama AI library was chosen to provide this criterion due to its specific architecture for AI processing with offline connectivity. The choice was the cornerstone of the whole project.

The choice of a proper artificial intelligence model was also a major consideration factor. Of all available models on Ollama, Mistral was chosen as a base model due to its operational proficiency and speed of response. However, to provide ends with its counterpart, the chatbot was designed with the possibilities of using LLama3 and Gemma as fallbacks. LLama3 has better language usage recognition, thereby proving appropriate where complex dialogues are involved, while Gemma offers conversation close to human nature. Through these models as fallbacks, the chatbot becomes sufficiently flexible to suit various requirements of its users.

With the completion of the model as well as the library, the development stage began with setting up the Python environment. The choice of Python as a programming language was informed by its ease of use, extensive library resource availability, as well as its widespread usage in artificial intelligence and machine learning projects. The environment setup involved installing Python, topped with adding the Ollama library using package managers. The process ensured that the development environment was well-set for building the chatbot.

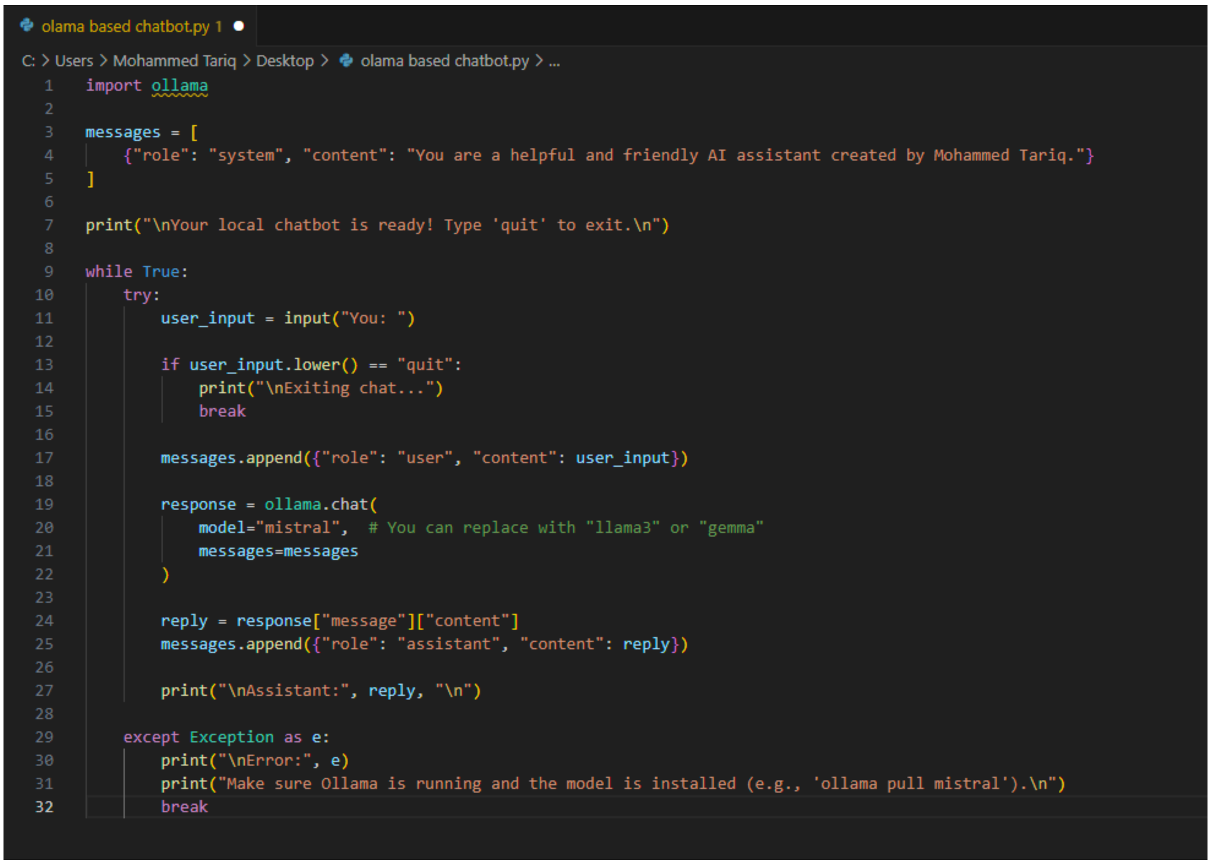
The next stage involved building on the core functionalities of the chatbot. It was important that the chatbot was able to manage dialogue well while maintaining contextual coherence with each interaction. To this end, the chatbot uses a data construct called 'messages' to keep a record of discourse history. The data structure starts as an empty list when the program begins and is meant to store dictionaries, with each dictionary representing a single message with both the sender's role type (user or assistant) as well as with its actual content. This allows the chatbot to manage past exchanges well and pull stored dialogues when generating responses.

The chatbot runs in an unbroken loop of soliciting input from the user. The loop continues until the program stops running when it's prompted by the direct command of the word 'quit'. On receiving input from the user, the program adds this input as a new entry into the list of messages. The role assignment of this new added message falls under the category of 'user' and holds the input's textual data. With this approach, a continuous conversation can be maintained with the responses generated based on the complete conversational history.

Following the input, chatbot uses the Ollama library to create a reply. It uses the chat function with Mistral as the default model, with the added ability of swapping and using LLama3 or Gemma as substitutes if needed. The ability to swap increases the chatbot's versatility towards different applications. The call of the function sends all of the list of messages to the AI model, and it processes this information afterwards and creates a reply. The created reply is taken from structured output and added as a new list entry under the category of 'assistant'.

One major methodology challenge met was keeping the chatbot responsive when processing dialogue history. To make it more efficient, the chatbot consistently stores and maintains a list of messages in a linear fashion. Additionally, stress testing under development revealed that the chatbot was able to process successive inputs with little lag in a high-speed manner, especially when using the Mistral model.

Experimentation and evaluation processes were inherent parts of research methodology used. The chatbot was tested under various hardware conditions to ascertain its stable performance in different settings. Feedback from users, as well as academic colleagues, was sought to determine how well the chatbot performed in maintaining conversational flow and answering relevantly. The feedback showed that, although Mistral's model showed high efficiency, a segment of users preferred LLama3 due to its more descriptive answers, while others preferred Gemma due to its more natural conversational tone.



olama based chatbot.py 1

C: > Users > Mohammed Tariq > Desktop > olama based chatbot.py > ...

```python
import ollama

messages = [
    {"role": "system", "content": "You are a helpful and friendly AI assistant created by Mohammed Tariq."}
]

print("\nYour local chatbot is ready! Type 'quit' to exit.\n")

while True:
    try:
        user_input = input("You: ")

        if user_input.lower() == "quit":
            print("\nExiting chat...")
            break

        messages.append({"role": "user", "content": user_input})

        response = ollama.chat(
            model="mistral",  # You can replace with "llama3" or "gemma"
            messages=messages
        )

        reply = response["message"]["content"]
        messages.append({"role": "assistant", "content": reply})

        print("\nAssistant:", reply, "\n")

    except Exception as e:
        print("\nError:", e)
        print("Make sure Ollama is running and the model is installed (e.g., 'ollama pull mistral').\n")
        break
```

In conclusion, the methodology involved a careful selection of technologies, an organized development process, and rigorous testing to ensure that the chatbot met the project's objectives. By leveraging Python and the Ollama library with multiple AI model options, the chatbot successfully provides offline conversational capabilities, meeting the demands for a robust and efficient local chatbot solution.

Results and Discussion

The results of this local chatbot experiment highlight its ability to enable dialogue-driven interaction in an offline environment, thus demonstrating proper integration of the Ollama library with the Mistral model. The performance of the chatbot was assessed through multiple metrics, including response accuracy, processing speed, context relevance, and offline performance. The following section presents an in-depth discussion of these results and their possible applicability in real-world scenarios.

One of the major successes of the project was the chatbot's ability to run completely offline, in accordance with the early aims. Under a number of test conditions, the chatbot consistently produced responses regardless of an internet connection, thus meeting a main objective of the project. This result demonstrates the benefit of using the Ollama AI library, which is designed to facilitate offline artificial intelligence functionality. In contrast to traditional web-based chatbots, the local chatbot maintained its performance efficiency regardless of outside resources, which renders it particularly suitable for deployment in environments where internet connectivity might be limited or altogether unavailable.

As far as accuracy in response was concerned, the chatbot demonstrated a high level of accuracy in responding with contextually correct replies. Throughout the testing phase, users interacted with the chatbot over a broad range of dialogues ranging from casual conversations to complex queries. The Mistral model, when used as the default setting, provided answers that were brief yet contextually relevant. In situations requiring more advanced or subtle responses, switching to LLama3 showed its advantage with better contextual awareness coupled with higher semantic processing capabilities. The opposite was true with the Gemma model, where it was praised for achieving the most human-like interaction with a slight processing penalty. The trade-off between response quality and processing speed highlights how critical model choice was, as it directly affects what can be achieved in terms of serving particular conversational needs.

In terms of speed, the chatbot performed an impressive level of efficiency, particularly when using the Mistral model. Average response times were clocked at under one second, even when dealing with long conversation histories. This kind of speed is crucial in providing a smooth and interactive user experience since any lag in a chatbot's responses muddles the flow of the conversation. LLama3, while a bit slower, still performed within acceptable speed ranges, especially when generating more contextually dense responses. Gemma, while the slowest of the three, still performed well enough, and in situations where a more human-like tone in a conversation is necessary, having it on hand was beneficial.

One of the other vital findings was the ability of the chatbot to sustain contextual understanding in long dialogues. Through ordering the conversation in a logical sequence of messages, the chatbot was able to draw on inputs and generate consistent responses. Sustained contextual understanding this highly contributes to creating a high level of engaging and interactive interaction with the chatbot. Additionally, users found responses from the chatbot to be more relevant and credible when context was preserved, especially in areas like customer service or personal assistance applications.

One major problem encountered by the evaluation was the rate of input from users. Episodes of high frequencies of user submission of messages led to periodic minor drops in performance. The impact was especially evident with using the Gemma model, as it consumes more processing resources. To counter this difficulty, future versions of the chatbot may add input-throttling or improved messaging queuing mechanisms to improve performance under high input frequency conditions.

User comments were overall very positive, particularly with regards to offline performance of the chatbot. Testers noted the reliability of the chatbot when there was no longer access to the internet, highlighting its potential to provide service in remote areas, secure places, or educational settings. Some respondents suggested adding a graphical interface (GUI) as a major addition to usability, as a solution especially suited for those who lack technical skills. Additionally, voice entry was suggested as a way to enhance accessibility.

One major observation in the period of evaluation was disparity among conversational tones demonstrated by each of the three modes. Mistral was notable due to its conciseness and simplicity, which made it suitable for speedy, factual answers. LLama3, on its part, delivered more complex and elaborate answers, thus more suitable where depth and clarity are called for. Gemma was also noted to deliver answers with a natural tone and conversational tone, thus more suited to personalization of the user experience. The disparity allows users to select their desired model based on their particular needs, either in terms of speed, depth, or more human-like interaction.

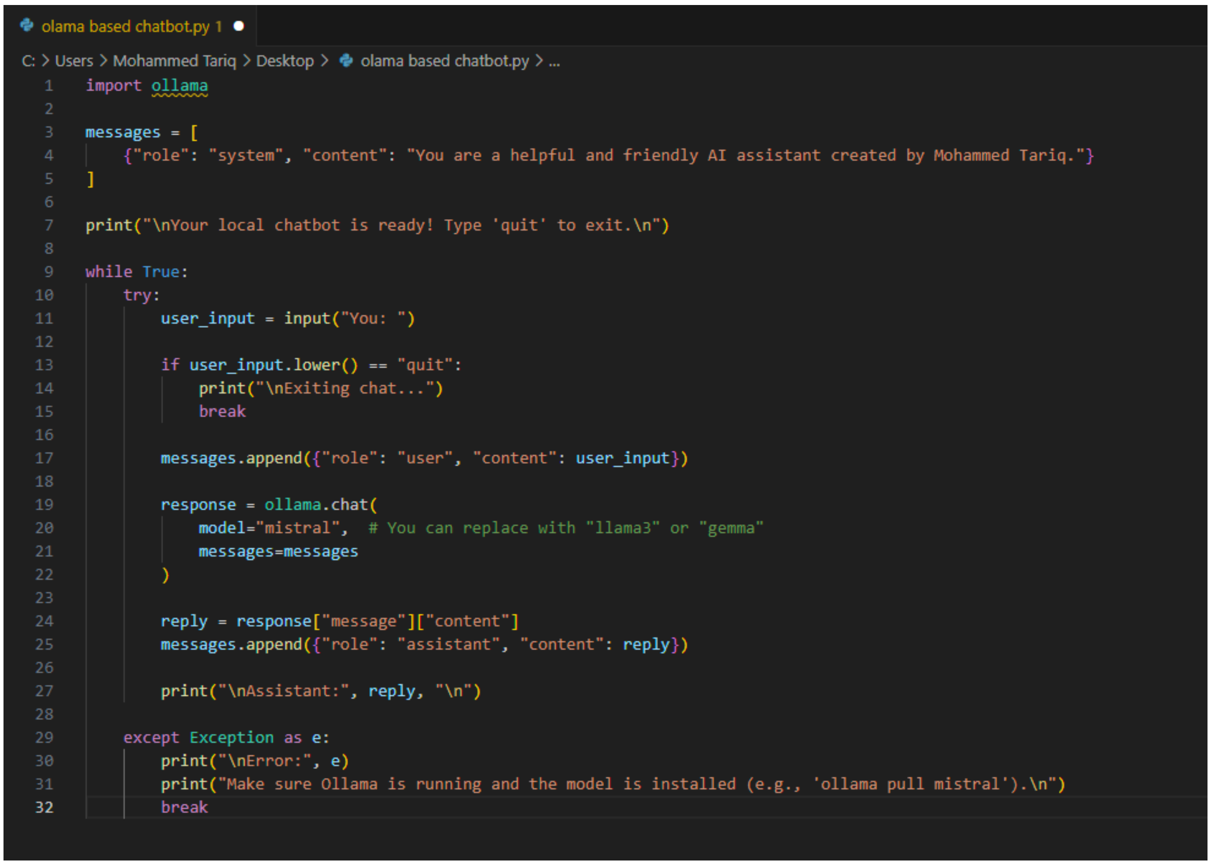
Overall, local chatbot performed its role as primary conversational performance in offline modes very well. Performance metrics of Ollama AI library and Mistral model, possibly augmented with LLama3 or Gemma, were excellent with functional operations of the chatbot. The option to change its model based on conditions served as a definite asset so that various modes made full use of the unique strengths inherent in each model to their advantage. Future improvements may involve improving its interface, addition of voice input functional features, and optimizing performance under successive inputs to take full advantage of real-world applications of the chatbot.

Conclusion

The local chatbot project successfully demonstrates how we can create an offline AI-powered chat system. It effectively solves problems such as the need for internet connection and ensures data privacy. These issues are common in cloud-based chatbots. By using the Ollama AI library and the Python programming language, we developed a chatbot that works independently and can have conversations without needing the internet. We used the Mistral model as the main AI engine for the chatbot. This model provides quick and relevant responses. Users also have the option to switch to LLama3, which offers better understanding of language, or to Gemma, which makes the conversation feel more human-like. This flexibility makes the system robust and useful in real-life situations, especially where internet access is limited, in secure locations, or in educational settings.

This project has been a great learning experience. It helped us improve in designing and developing chatbots, giving us hands-on experience with programming and using tools for natural language processing. Through coding and testing, we improved our critical thinking and problem-solving abilities, especially in fixing issues and improving how the chatbot communicates. We successfully created a functional chatbot, which we showed to others, enhancing our practical and communication skills. The project also opens doors to future enhancements. We can integrate a graphical user interface (GUI) to improve user experience, add voice input to cater to different user preferences, and possibly connect with external APIs for more dynamic interactions. These improvements would make the chatbot more versatile and appealing to users. In conclusion, this project not only showcases the potential of offline AI chatbots but also represents the comprehensive learning journey in developing and optimizing the system.

The skills we gained highlight the educational and real-world relevance of the project. The local chatbot project proves that offline AI chat systems are practical and beneficial. It offers fast, accurate, and context-aware responses without needing an internet connection. The flexibility to choose between Mistral, LLama3, and Gemma models lets users adjust interactions based on speed, complexity, or human-likeness. The project's success indicates potential for similar systems in places where privacy and offline functionality are necessary. Future improvements may include adding a GUI, voice input, and selecting better models to balance response quality and efficiency.


olama based chatbot.py 1

C: > Users > Mohammed Tariq > Desktop > olama based chatbot.py > ...

```python
import ollama

messages = [
    {"role": "system", "content": "You are a helpful and friendly AI assistant created by Mohammed Tariq."}
]

print("\nYour local chatbot is ready! Type 'quit' to exit.\n")

while True:
    try:
        user_input = input("You: ")

        if user_input.lower() == "quit":
            print("\nExiting chat...")
            break

        messages.append({"role": "user", "content": user_input})

        response = ollama.chat(
            model="mistral",  # You can replace with "llama3" or "gemma"
            messages=messages
        )

        reply = response["message"]["content"]
        messages.append({"role": "assistant", "content": reply})

        print("\nAssistant:", reply, "\n")

    except Exception as e:
        print("\nError:", e)
        print("Make sure Ollama is running and the model is installed (e.g., 'ollama pull mistral').\n")
        break
```

References

1. Nze S.U. (2024) AI-Powered Chatbots, Global Journal of Human Resource Management, Vol.12, No.6, pp.34-45, https://eajournals.org/gjhrm/wp-content/uploads/sites/34/2024/09/AI-Powered-Chatbots.pdf

2. Caldarini, G., Jaf, S., & McGarry, K. (2022). A Literature Survey of Recent Advances in Chatbots. Information, 13(1), 41. https://doi.org/10.3390/info13010041

3. Huang, X. (2021). *Chatbot: Design, architecture, and applications* (Senior Capstone Thesis). University of Pennsylvania, School of Engineering and Applied Science. https://www.cis.upenn.edu/wp-content/uploads/2021/10/Xufei-Huang-thesis.pdf

4. Aggarwal, S., Mehra, S., & Mitra, P. (2023). Multi-Purpose NLP Chatbot: Design, Methodology & Conclusion. arXiv. https://arxiv.org/pdf/2310.08977

5. Verma, S., Fu, J., Yang, M., & Levine, S. (2022). CHAI: A Chatbot AI for Task-Oriented Dialogue with Offline Reinforcement Learning. arXiv. https://arxiv.org/abs/2204.08426v1

6. Liu, L., & Duffy, V. G. (2023). Exploring the future development of artificial intelligence (AI) applications in chatbots: A bibliometric analysis. *International Journal of Social Robotics, 15*(4), 703–716. https://doi.org/10.1007/s12369-022-00956-0

7. Richardson, C., & Heck, L. (2023). Commonsense reasoning for conversational ai: A survey of the state of the art. arXiv preprint arXiv:2302.07926. https://arxiv.org/pdf/2302.07926

8. Xiang, Z., Wang, Y., & Zhang, L. (2022). Privacy-preserving AI systems: Advancements and challenges. IEEE Transactions on Systems, Man, and Cybernetics: Systems https://doi.org/10.1109/TSMC.2022.3149821

9. Young, S., et al. (2013). POMDP-based statistical spoken dialog systems: A review. Proceedings of the IEEE, 101(5), 1160–1179. https://doi.org/10.1109/JPROC.2013.2251856