Mental Health Assistance Chatbot

FINAL REPORT

Project-Based Learning and Technology (PBT205)

Prepared by:

1. Smriti Parajuli (IMDS3000025)

2. Gloria Hawkins-Roberts (MDS2000662)

Submitted to:

Ranpreet Kaur, Lecturer, BSE-AI

Media Design School

August 29th, 2024

Abstract:

This project explores the development process of the Mental Health Assistance Chatbot through the utilisation of a large language model (LLM) and natural language processing (NLP) techniques. The chatbot is trained on the Mental Health Conversations Dataset (*Amod/Mental_Health_Counseling_Conversations · Datasets at Hugging Face*, 2001), and aims to provide support and guidance for those needing mental health assistance. By leveraging LLM capabilities, the chatbot understands users' mental health concerns and is able to respond to them appropriately. This project aims to demonstrate the potential of AI in providing accessible mental health care support, offering an innovative solution that addresses the increasing demand for mental health assistance.

1. Introduction:

Mental health is a critical aspect of overall well-being, yet to many mental health support access remains limited. For example, over 66% of Americans diagnosed with a mental health condition were unable to access mental health care in 2021 due to financial restriction, social stigma, limited access, and location/ accessibility (Chatterjee, 2023). This project aims to bridge that gap by developing a mental health assistance chatbot that leverages large language models (LLMs) and natural language processing (NLP) techniques. The chatbot is designed to provide conversational support and guidance to users, offering an accessible and anonymous platform for discussing mental health concerns.

The dataset used for this project is a collection of over 3,500 mental health conversation entries, structured in a JSON format (*Amod/Mental_Health_Counseling_Conversations · Datasets at Hugging Face*, 2001). It was sourced from two online counselling and therapy platforms, where individuals sought advice on a wide range of mental health topics, and responses were provided by qualified psychologists. This dataset is specifically designed for fine-tuning language models to enhance their ability to deliver accurate and empathetic mental health advice (*Amod/Mental_Health_Counseling_Conversations · Datasets at Hugging Face*, 2001). Given the sensitive nature of the content, all personal information in the dataset is anonymized, ensuring that no personally identifiable information is included (*Amod/Mental_Health_Counseling_Conversations · Datasets at Hugging Face*, 2001). Additionally, the dataset was meticulously cleaned to retain only the relevant conversation data, and it does not contain any additional annotations (*Amod/Mental_Health_Counseling_Conversations · Datasets at Hugging Face*, 2001).

The motivation for this project stems from the ever-growing need and demand for mental health support services, along with the challenges associated with meeting this demand. Traditional mental health services are often constrained by factors such as availability, cost, and stigma. By incorporating AI and NLP, this project aims to create a scalable and accessible solution that can provide immediate support to those in need.

NLP and LLMs are well-suited for this project due to their ability to understand and generate human-like text. NLP techniques enable the chatbot to process and analyse the users inputs, while the LLM's advanced capabilities allow it to generate contextually appropriate responses. The use of an LLM ensures that the chatbot can engage in meaningful and empathetic conversations, making it a valuable tool for mental health assistance and the Mental Health Assistance Chatbot.

In summary, this project seeks to harness the power of AI to create a mental health chatbot that can assist users in navigating their mental health challenges. By utilising a suitable and specific dataset along with NLP and LLM techniques, the chatbot aims to provide accessible, supportive, and empathetic mental health assistance to a broad audience.

2. Projects Significance and Need:

The significance of this project lies in its potential to address a critical gap in mental health care accessibility. Mental health issues are pervasive, affecting 1 in 8 people worldwide (World Health Organization: WHO, 2022), yet many people struggle to access the support they need due to various barriers, including cost, location, availability of professionals, and societal stigma. This project's AI-powered mental health assistance chatbot offers a scalable and accessible solution that can help bridge this gap.

The need for this type of solution is evident in the increasing demand for mental health services. Traditional approaches often fall short in meeting this demand, especially in underserved communities where resources are limited. For example, the average therapy session in New Zealand can cost between $150 and 200 dollars per hour (Nz, 2022), this is inaccessible to many due to financial restraints. Figure 1 further demonstrates the limited availability, in New Zealand there are only around 10 psychologists per 100,000 people (Nz, 2022). By providing an always-available, anonymous, and cost-effective platform for mental health support, this chatbot can make a meaningful difference in the lives of individuals who might otherwise go without help.

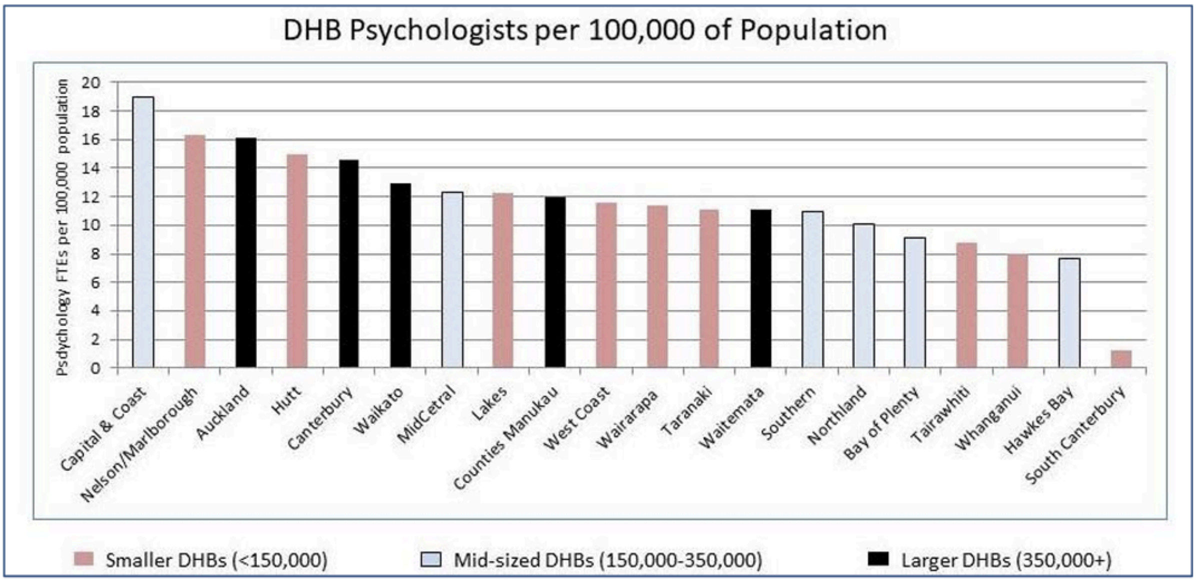

Figure 1. Source: DHB Psychology Professional Leaders/Advisors. Also reported in APEX Psychologists Newsletter, Sept 2016. (Nz, 2022).

Moreover, the significance of this project extends beyond just providing immediate support. The chatbot can serve as a first line of intervention, offering users a way to express their thoughts and emotions in a safe space, potentially reducing the severity of mental health crises before they escalate. For those hesitant to seek professional help, the chatbot can act as a bridge, encouraging them to take the next step in their mental health journey.

This project also highlights the ethical considerations of deploying AI in sensitive areas such as mental health. By focusing on empathetic communication and maintaining user privacy, the project demonstrates how AI can be responsibly used to support mental well-being. The insights gained from this case study can inform future developments in AI-driven mental health solutions, making it a significant contribution to both the AI and mental health fields.

In conclusion, this project addresses a pressing societal need by providing an accessible, scalable, and empathetic mental health assistance tool. Its significance lies in its potential to improve mental health outcomes for a wide range of individuals, particularly those who face barriers to traditional mental health services.

3. Aims and objectives:

The primary aim of this project is to develop an AI-driven mental health assistance chatbot through python that leverages the capabilities of the OpenAI GPT-3.5 Turbo model, NLP, and speech recognition technologies. The chatbot is designed to engage users in speech-based conversations, providing support and guidance in a counselling scenario.

To achieve this aim, the following objectives have been established:

1. **Integration of OpenAI GPT-3.5 Turbo Model:**
   - Utilisation of the fine-tuned OpenAI GPT-3.5 Turbo model (ft:gpt-3.5-turbo-0125) to generate contextually relevant and empathetic responses based on the user's input. The model should simulate a counsellor-like interaction, guiding the user through the conversation and offering supportive responses.
2. **Natural Language Processing (NLP) Implementation:**
   - The application of NLP techniques to analyse and understand the user's input effectively, ensuring that the chatbot can comprehend various mental health topics and respond appropriately. NLP will enhance the chatbot's ability to interpret context, making the interaction more meaningful and effective.
3. **Speech Recognition Implementation:**
   - The incorporation of speech recognition capabilities by using the python speech_recognition library that allows users to interact with the chatbot through their voice. This sits in line with objectives to improve accessibility, this means that those who cannot type can easily just use their voice to communicate with the chatbot. This further propels the chatbot to mimic a real life conversational experience.The system should accurately capture and process the user's spoken input, ensuring a seamless conversational experience.
4. **Text-to-Speech (TTS) Integration:**
   - Implement TTS functionality using the python pyttsx3 library to vocalise the AI's responses. This will enhance the user experience by providing a natural

and interactive dialogue flow, similar to a real counselling session. This aligns with the user's use of speech to communicate with the chatbot and further enhances the experience of mimicking counselling.

5. **GUI Development:**
    - The development of a user-friendly, however simple, graphical interface using Tkinter to display the conversation between the user and the chatbot. The interface allows the user to view the ongoing conversation as text, including both user input and AI responses.
6. **Conversation Saving Feature:**
    - Creating a feature that enables users to save/ download their conversation history to a text file. This will allow users to keep a record of their sessions, which they can review later or may find useful for therapy or reflection later on.
7. **Speech-Based Conversational Flow:**
    - Ensuring that the chatbot can maintain a smooth and engaging speech-based conversational flow by recognizing user input, generating appropriate responses, and prompting further conversational input from the user.
8. **Accessibility and User Experience:**
    - A focus on creating an accessible and intuitive user experience, ensuring that the chatbot can effectively assist users in navigating their mental health concerns through a natural and supportive conversation.

The overarching goal of this project is to provide users with a supportive and accessible mental health tool that mimics real counsellor interactions using advanced AI, NLP, speech recognition, and TTS technologies. This goal is supported by these aims and objectives.

4. Methodology and Approach:

4.1 Problem Domain

The problem domain of this project lies within the intersection of mental health assistance and artificial intelligence. Specifically, this project aims to develop a mental health chatbot that can assist users through conversational interactions, providing support and guidance. Mental health issues are prevalent and require accessible, scalable, and empathetic solutions.

Traditional methods of delivering mental health support often face barriers such as stigma, cost, and limited availability of professionals. The chatbot aims to address these challenges by offering a speech-based, AI-driven solution that users can interact with in a natural and supportive manner.

4.2 OpenAI GPT-3.5 Turbo Model:

The implementation of the OpenAI GPT-3.5 Turbo model for this project involved several key steps to ensure that it could effectively simulate counsellor-like interactions. First, the decision to use GPT-3.5 Turbo was made after evaluating other language models, such as GPT-2, which struggled with maintaining context and delivering appropriate responses in sensitive mental health conversations. GPT-3.5 Turbo, in contrast, demonstrated superior performance in understanding and generating responses that align with the user's emotional and conversational needs.

The process began with the collection and preparation of the dataset, which included over 3,500 conversation entries between individuals and mental health professionals (*Amod/Mental_Health_Counseling_Conversations · Datasets at Hugging Face*, 2001). The dataset was meticulously cleaned and formatted into a structure that the GPT-3.5 Turbo model could understand, specifically using a JSON format with dialogue pairs representing user input and counsellor responses (*Amod/Mental_Health_Counseling_Conversations · Datasets at Hugging Face*, 2001).

Next, the dataset was uploaded to the OpenAI platform for fine-tuning. The fine-tuning process involved training the GPT-3.5 Turbo model on this specific dataset to help it learn the nuances of mental health dialogues. This training allowed the model to adapt its responses based on the context, ensuring that it could generate replies that are both empathetic and contextually appropriate.

Once the model was fine-tuned, it was integrated into the chatbot application. During this integration, the fine-tuned model was tested with various user inputs to ensure it could handle a wide range of mental health-related queries effectively. Parameters such as temperature and max tokens were fine-tuned to balance the length and creativity of the responses, ensuring that the chatbot could engage in meaningful and supportive conversations with users. This step-by-step implementation allowed the GPT-3.5 Turbo model to be optimised for delivering high-quality mental health support through the chatbot.

4.3 Natural Language Processing (NLP):

NLP techniques play a crucial role in the development of this chatbot, particularly in the preprocessing of text data. NLP is employed to clean and standardise the data, enabling the GPT-3.5 Turbo model to better understand user inputs. The preprocessing steps included lowercasing, punctuation removal, and whitespace cleanup, all of which contribute to making the text data uniform and easier for the model to analyse. By removing unnecessary noise such as punctuation and extra spaces, the text becomes more consistent, which in turn improves the model's ability to generate coherent and accurate responses. This data cleaning process ensures that the model is trained on high-quality data, which is critical for achieving optimal performance in the chatbot.

4.4 Speech Recognition (speech_recognition Library):

To make the chatbot more interactive and accessible, the speech_recognition library is used to capture user input through voice. This allows users to engage in real-time, speech-based conversations, making the interaction more natural and user-friendly. The library's ability to handle ambient noise is particularly important, as it ensures that the speech recognition system performs well in various environments, whether the user is in a quiet room or a noisy setting. The ambient noise adjustment feature helps improve the accuracy of the speech recognition process, ensuring that the chatbot accurately captures the user's spoken input, which is then processed to generate an appropriate response.

4.5 Text-to-Speech (TTS) (pyttsx3 Library):

To further enhance the user experience, the **pyttsx3 library** is employed to vocalise the chatbot's responses. This adds an auditory element to the interaction, making the conversation more dynamic, engaging, and realistic. By converting the AI-generated text into speech, the chatbot simulates a more human-like interaction, which is particularly valuable in a mental health context where a supportive and empathetic tone is crucial. The pyttsx3 library allows for fine-tuning of the speech rate and volume, ensuring that the responses are delivered clearly and naturally, enhancing the overall user experience.

4.6 Tkinter for GUI:

The Tkinter library is used to create a simple and user-friendly graphical interface (GUI) for the chatbot. The GUI provides a platform where the conversation between the user and the chatbot is displayed, making the interaction more accessible. Users can interact with the chatbot visually and audibly, and the interface includes features such as the ability to download the conversation for future reference. Tkinter's flexibility allows for the creation of a clean and organised interface that is easy for users to navigate, ensuring a smooth user experience.

4.7 Data Preprocessing

The initial step in preparing the dataset for fine-tuning the GPT-3.5 Turbo model involves thorough data preprocessing. This step is crucial for ensuring that the data is clean, consistent, and ready for training. Several key actions were taken during this phase:

1. **Lowercasing:**
    - Text data often includes variations in case (e.g., "Happy" vs. "happy"), which can lead to inconsistencies during model training. By converting all text to lowercase, uniformity is ensured, reducing the likelihood of the model interpreting the same word differently based on case. This was implemented using Python's string manipulation functions to convert the text in both the "Context" and "Response" columns to lowercase.
2. **Punctuation Removal:**
    - Punctuation, while important in natural language, can introduce noise in training data for certain NLP tasks. For this chatbot, where the focus is on understanding the content of conversations rather than grammatical correctness, removing punctuation helps the model concentrate on the actual words being used. The str.translate() function was employed to strip away symbols like periods, commas, and question marks.
3. **Whitespace Cleanup:**
    - Inconsistent spacing, such as multiple spaces between words, can add unnecessary complexity to the data. Cleaning up whitespace ensures that the

text is uniformly spaced, which helps in maintaining data quality. Regular expressions (re.sub()) were used to replace multiple spaces with a single space and to trim leading or trailing whitespace from the text.

4. **Duplicate Removal:**
   - The 750 duplicates in the dataset can bias the model during training, causing it to overfit on certain patterns that are overrepresented. Removing duplicates ensures that the model learns from a diverse set of examples, enhancing its generalisation ability. The drop_duplicates() function in pandas was used to identify and remove any duplicate rows in the dataset.

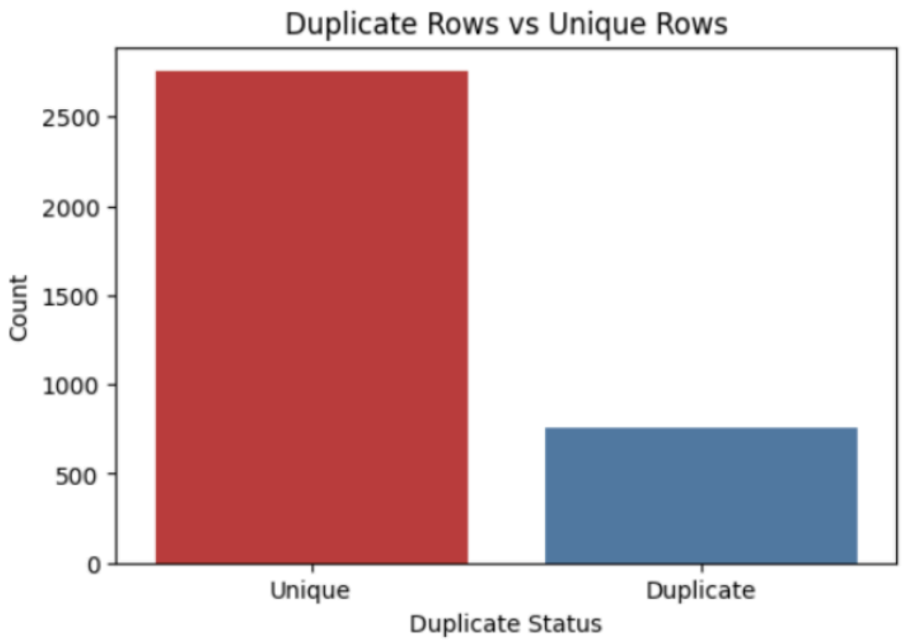

Figure 2. A graph of the number of unique vs duplicate rows in the dataset, generated via preprocessing code.

5. **Text Length Analysis:**
   - Analysing the length of the text in both the "Context" and "Response" columns provides insights into the structure of the conversations. By understanding the distribution of word counts, it is ensured that the model is trained on a variety of conversation lengths, helping it handle both short and long dialogues effectively. Python's len() function was used to calculate word counts, and histograms were plotted using Seaborn to visualise the distribution. This showed us that the responses were longer in length within the dataset (figure 2)

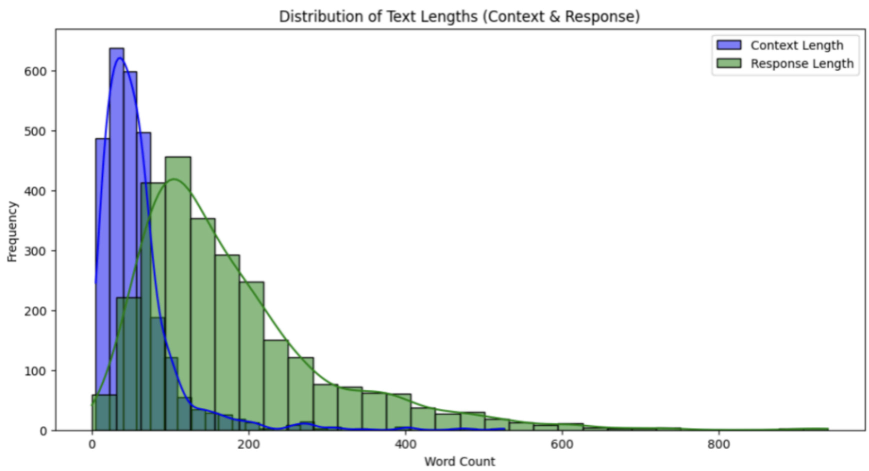

Figure 3. A graph of different text lengths within the dataset generated by preprocessing code.

6. **Visualisation:**
   - Visualising the distribution of word counts helps identify potential outliers or patterns in the dataset that could impact the model's performance. For example, if most responses are very short, the model might need adjustments to handle longer, more complex inputs. Histograms and KDE plots were generated using Seaborn to show the frequency distribution of word counts for both context and response texts.

4.8 Fine-Tuning Process

1. **Formatting the Dataset:**
   - The GPT-3.5 Turbo model expects input in a specific format, particularly when fine-tuning for conversation tasks. The data was structured as a series of interactions between a user and an assistant (in this case, a counsellor).
2. **JSON Format:**
   - OpenAI requires the dataset to be saved in JSON Lines (JSONL) format for fine-tuning. This format allows each conversation pair to be stored as a separate line, making it easier to process large datasets line by line. The formatted data was found as a .json file.

3. **Uploading the Dataset:**

   - Before fine-tuning can begin, the dataset must be uploaded to OpenAI's platform. This step makes the data accessible to the fine-tuning process. The dataset was uploaded using the openai.File.create() API call, which returns a file ID used to reference the dataset in the fine-tuning job.
4. **Starting the Fine-Tuning Job:**
   - Fine-tuning adapts a pre-trained model (like GPT-3.5 Turbo) to a specific domain or task. In this case, fine-tuning ensures that the model generates more relevant and contextually appropriate responses for mental health conversations. The fine-tuning job was initiated using the openai.FineTuningJob.create() API call, specifying the uploaded file and base model (GPT-3.5 Turbo) to be fine-tuned.
5. **Monitoring the Fine-Tuning Process:**
   - It was important to monitor the process to ensure that it completes successfully. If any issues arised (e.g., training fails or gets stuck), they needed to be addressed promptly. A loop was created using the openai.FineTuningJob.retrieve() API call to periodically check the status of the fine-tuning job. The loop waits 60 seconds between checks to avoid overwhelming the API with requests.
6. **Post-Fine-Tuning:**
   - After fine-tuning is complete, the resulting model must be tested to ensure it meets the desired performance criteria. The fine-tuned model ID is retrieved and used in subsequent chatbot interactions. Sample inputs were tested to verify the quality of the responses, and adjustments to the model's temperature and max token settings were made as needed.
   -

4.9 Speech Recognition and Response Handling

1. **Speech Recognition:**
   - To create a natural conversational experience where users can speak their inputs instead of typing, the speech_recognition library was used. This library captures audio input from the user's microphone and converts it into text using the recognize_google() method. Ambient noise adjustment was applied to

improve recognition accuracy in various environments, ensuring the chatbot captures the user's spoken input effectively.

2. **Handling User Input:**
   - Proper handling of user input is crucial, especially in situations where speech recognition may fail (e.g., due to unclear speech or background noise). Error handling mechanisms were implemented to detect failures in speech recognition. If an error occurs, the chatbot prompts the user to repeat their input, ensuring that the conversation can continue without disruption.
3. **Generating AI Responses:**
   - After capturing the user's input, the chatbot needs to generate a relevant response based on the conversation context. This is where the fine-tuned model is utilised.

5. Chatbot Evaluation:

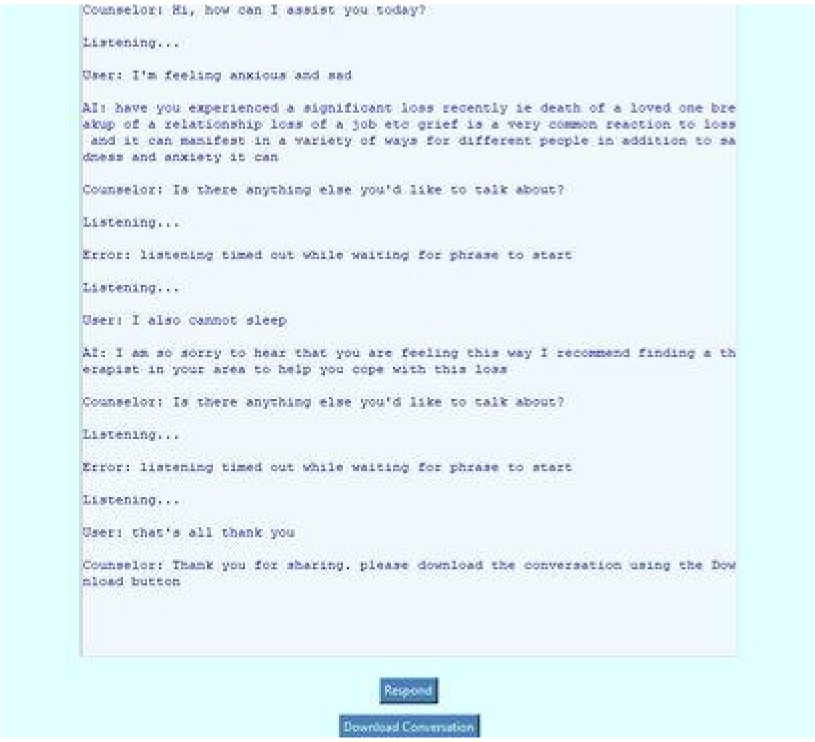

Figure 4. GUI output of the Mental Health Assistance Chatbot.

Figure 5. GUI output of the Mental Health Assistance Chatbot.

Figure 6. GUI output of the Mental Health Assistance Chatbot.

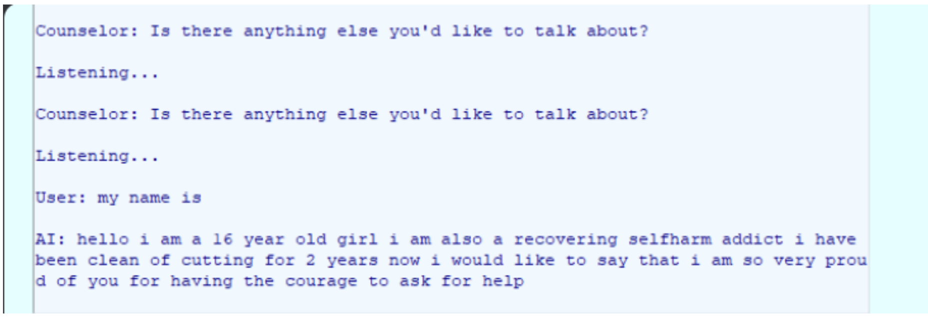

Figure 7. GUI output of the Mental Health Assistance Chatbot.

5.1 Areas for Improvement:

Speech Recognition Reliability: As observed in the conversation logs, the chatbot sometimes fails to recognize or process the user's input , resulting in errors such as "listening timed out while waiting for a phrase to start." (see Figure 4). This issue disrupts the flow of conversation and can frustrate users. To improve this, the timeout settings in the speech recognition library could be fine-tuned, or the system could prompt the user more effectively when input isn't detected.

Response Quality: While the chatbot generally provides contextually appropriate responses (figure 5) , there are instances where the responses are not as coherent or helpful as expected, especially when the user is not specific or clear (Figure 4). For example, the chatbot might generate a generic response that doesn’t fully address the user's concern (see Figure 4). Improving the fine-tuning process or incorporating additional context could help in generating more precise and relevant responses. There are instances where the chatbot responds well if the user is clear and specific

Handling of Silence or Errors: The chatbot currently does not handle cases where the user might remain silent or when an error occurs during speech recognition smoothly. Enhancing the error handling mechanism to provide more informative prompts or suggestions when the system fails to capture input would improve the user experience. The chatbot also is unable to process personal information as seen in figure 7. This prompts the need for more robust error handling.

Consistency in Interaction: The chatbot sometimes fails to respond in a timely manner or generates responses that do not fully align with the user's input. This can break the flow of conversation and reduce the overall effectiveness of the tool (see Figure 4). Adding more robust testing and potentially training the model on a broader dataset could help address these inconsistencies.

Complexity of Responses: The chatbot currently handles relatively straightforward interactions, but it struggles with more complex or nuanced mental health queries such as specific mental illnesses. The chatbot is primarily best at addressing basic queries such as anxiety at work, and sadness at home, grief of a loved one or pet etc. Extending the model's capabilities by incorporating additional data or using more advanced AI techniques could enable it to handle a wider range of conversation topics and provide more in-depth support. Further datasets would also be needed to implement more complex responses.

5.2 Constraints and Challenges:

API Costs: One of the significant constraints is the cost associated with using the OpenAI GPT-3.5 Turbo API. While the model is highly effective, continuous use, especially for a high-volume application like a chatbot, can lead to substantial costs. This could be a limiting factor for long-term deployment, especially for projects with limited budgets. Exploring cost-effective alternatives, optimising API usage, or securing funding/ budgeting planning could help mitigate this issue. This was very limiting for our testing process and fine-tuning, it meant that we were unable to explore the potential of the model enough due to the API costs.

Dataset Size and Quality: The dataset used for training and fine-tuning the model, while substantial, may still limit the chatbot's ability to handle a wide variety of mental health scenarios. Expanding the dataset to include more diverse conversations and ensuring it covers a broader range of topics could enhance the model's performance. Additionally, larger datasets often require more computational resources and longer training times, which could be a challenge depending on available resources.

Infrastructure Requirements: Running a chatbot that uses advanced AI models like GPT-3.5 Turbo requires robust infrastructure, particularly in terms of computing power and network stability. These requirements can add to the operational costs and may limit the scalability of the solution.

5.3 Future Enhancements:

Enhanced Speech Recognition: Improving the accuracy and reliability of speech recognition will make the interaction smoother and reduce instances of errors or timeouts.

Context-Aware Responses: Enhancing the model's ability to understand and maintain the context of longer or more complex conversations could lead to more meaningful interactions.

Error Handling: Developing more robust error handling mechanisms to guide the user through issues with speech recognition or other system failures.

Emotional Sentiment Analysis: Implementing sentiment analysis could allow the chatbot to adjust its tone and response based on the user's emotional state, making the interaction more empathetic and supportive.

Video Avatar for a More Engaging Interface: Instead of just a basic GUI, introducing a video avatar or a virtual assistant with facial expressions and gestures could make the interaction more engaging and personable. A video avatar could provide visual cues that enhance the user's sense of connection and support, making the chatbot feel more human-like. This feature could be particularly beneficial in making the mental health support experience more immersive and comforting.

Expanding language Support: Expanding different languages compatible with the chatbot to further propel accessibility, especially for non-english speakers.

6. Timeframe Estimation for Each Phase

1. Problem Domain Analysis (1-2 days)

- **Activities:** Define the problem domain, identify key challenges in mental health assistance, and outline how AI can address these challenges.

- **Expected Time:** 1-2 days
- **Actual Time:** 1 week

2. Data Collection and Preprocessing (3-4 days)

- **Activities:** Collect and preprocess the dataset, including steps like lowercasing, punctuation removal, whitespace cleanup, and duplicate removal. Perform text length analysis and visualise the data distribution.
- **Expected Time:** 3 days
- **Actual Time:** 2 weeks

3. Fine-Tuning the GPT-3.5 Turbo Model / also selecting a model (7 days)

- **Activities:** Format the dataset for fine-tuning, save it in JSONL format, upload it to the OpenAI platform, and start the fine-tuning process. Monitor the fine-tuning job and make adjustments as needed.
- **Expected Time:** 7 days
- **Actual Time:** 2 weeks

4. Integrating Speech Recognition and TTS (3-4 days)

- **Activities:** Implement the speech_recognition library to capture user input and integrate the pyttsx3 library for text-to-speech conversion. Test the accuracy and performance of these integrations in different environments.
- **Expected Time:** 3 days
- **Actual Time:** 1 days

5. GUI Development with Tkinter (3-4 days)

- **Activities:** Develop a user-friendly GUI using Tkinter, ensuring features like conversation display and download options are included. Create a clean and organised layout.
- **Expected Time:** 3 days
- **Actual Time:** 1 days

6. Testing and Debugging (4-5 days)

- **Activities:** Conduct comprehensive testing of the chatbot, including edge cases, error handling, and user experience evaluation. Make necessary adjustments based on testing feedback.
- **Expected Time:** 4 days
- **Actual Time:** 2 days

7. Final Adjustments and Documentation (2-3 days)

- **Activities:** Make final adjustments based on testing results, and prepare all necessary documentation, including the SRS document and project report.
- **Expected Time:** 2 days
- **Actual Time:** 4 days (2 days for each document)

7. Task lists

7.1 Collective:

1. **Select a problem domain to focus on**. This was expected to take 1-2 days but actually took a week as we could not find an appropriate dataset for most of the problem domains we chose.
2. **Select a dataset.** This was expected to take a few days but ended up taking us two weeks as we couldn't find something that coincided with the domain.
3. **Testing and debugging.** This involved testing the code, making sure the GUI functioned, and debugging the code. This was expected to take around 4 days and ended up taking 2 days.
4. **Report write up.** This was expected to take a couple of days and ended up taking the same amount of time we expected.
5. **SRS documentation**. This involved writing down our expectations and planning the project. This was expected to take a couple of days and ended up taking the same amount of time that we anticipated.
6. **GUI design.** This involved using Tkinter to implement a GUI. This was expected to take a couple of days and ended up taking around a day.

7.2 Smriti:

1. **Implementing the chosen model**. Making sure that the chosen model is implemented correctly in the code. This was expected to take a few days and ended up taking the same amount of time we expected.
2. **Fine tuning**. Adjusting the model. This was expected to take a few days and ended up taking around a week.
3. **Integrating speech recognition.** This involved adding speech recognition so that the user could talk instead of writing in their input. This was expected to take a couple of days and ended up only taking a day.
4. **Powerpoint.** Creating our final powerpoint slides. This was expected to take a day and only took a day.

8. Conclusion:

The development of the Mental Health Assistance Chatbot demonstrates the potential of AI-driven solutions in providing accessible mental health support. Through the integration of advanced technologies like the OpenAI GPT-3.5 Turbo model, NLP techniques, speech recognition, and text-to-speech capabilities, the chatbot simulates a conversational experience that can guide users through mental health challenges. The project effectively addressed the need for a scalable, anonymous, and cost-effective platform, making mental health care more accessible to those who might otherwise struggle to find help.

This project also highlighted the importance of careful dataset preparation, model fine-tuning, and user interface design in creating an effective and empathetic mental health tool. The chatbot's ability to engage users in speech-based conversations and provide meaningful responses demonstrates the strength of combining AI with mental health care.

However, there were also notable challenges, such as the limitations of speech recognition, the consistency of the responses, and the costs associated with running advanced AI models. These challenges emphasise the need for continuous improvement and refinement of the system to ensure it can meet the diverse needs of users. Future enhancements, such as improving the speech recognition reliability, handling more complex interactions, and incorporating emotional sentiment analysis, could further enhance the chatbot's capabilities.

Overall, this project serves as a valuable case study in the application of AI and NLP technologies in mental health care, offering insights into both the opportunities and challenges of deploying AI in sensitive domains. The Mental Health Assistance Chatbot has the potential to make a significant impact on mental health support, providing a valuable tool for individuals seeking help in navigating their mental health journey.

9. Resources:

1. *Amod/mental_health_counseling_conversations · Datasets at Hugging Face*. (2001, March 4). https://huggingface.co/datasets/Amod/mental_health_counseling_conversations

2. Chatterjee, R. (2023, December 13). Most Americans with mental health needs don't get treatment, report finds. *NPR*. https://www.npr.org/sections/health-shots/2023/12/13/1218953789/most-americans-with-mental-health-needs-dont-get-treatment-report-finds

3. Cull, G. (2024, January 11). Fine tuning AI models — a practical guide for beginners. *Medium*. https://medium.com/@garethcull/fine-tuning-ai-models-a-practical-guide-for-beginners-dc313b2e0f76

4. get.abhishekified. (2023a, August 10). *Finetuning LLM for Text Generation - Part 1* [Video]. YouTube. https://www.youtube.com/watch?v=5aXbwGwzpQ0

5. get.abhishekified. (2023b, August 10). *Finetuning LLM for Text Generation - Part 2* [Video]. YouTube. https://www.youtube.com/watch?v=MH8Lq_OzfoI

6. How Many Psychologists? (2017). In *Nzccp*. NZCCP.

7. Laledemir, Ç. (2024, May 13). Using OpenAI API with GPT-3.5-turbo in Python - Çağlar Laledemir - Medium. *Medium*. https://medium.com/@caglarlaledemir/openai-gpt-3-5-turbo-in-action-building-a-simple-chatbot-interface-41300696094b

8. Menon, P. (2023, March 10). Introduction to large language models and the Transformer architecture. *Medium*. https://rpradeepmenon.medium.com/introduction-to-large-language-models-and-the-transformer-architecture-534408ed7e61

9. Nantasenamat, C. (2023, August 11). Beginner's Guide to OpenAI API - Data Professor - Medium. *Medium*. https://medium.com/data-professor/beginners-guide-to-openai-api-a0420bc58ee5

10. Nz, C. (2022, September 28). *Counselling services: How to find a therapist*. Consumer NZ. https://www.consumer.org.nz/articles/how-to-find-a-therapist#:~:text=When%20we%20surveyed%20private%20providers,%24150%20to%20%24200%20per%20session

11. Schmid, P. (2024, January 23). How to Fine-Tune LLMs in 2024 with Hugging Face. *philschmid*. https://www.philschmid.de/fine-tune-llms-in-2024-with-trl

12. Thanumoorthy, P. (2024, February 19). Generative AI Large language models: basics, LLM abilities, and transformer architecture explained | Medium. *Medium*. https://medium.com/@padmathanumoorthy/large-language-models-understanding-basics-llm-abilities-and-transformer-architecture-model-7e01d064659e

13. World Health Organization: WHO. (2022, June 8). *Mental disorders*. https://www.who.int/news-room/fact-sheets/detail/mental-disorders

.