

©Copyright 2024

Sugam Jaiswal

Development of Personality Adaptive Conversational AI for Mental Health Therapy Using LLMs

Sugam Jaiswal

A thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science in Computer Science and Software Engineering

University of Washington

2024

Reading Committee:

Dong Si, Chair

Annuska Zolyomi

Muhammad Aurangzeb Ahmad

Program Authorized to Offer Degree:

Computer Science and Software Engineering

University of Washington

Abstract

Development of Personality Adaptive Conversational AI for Mental Health Therapy Using LLMs

Sugam Jaiswal

Chair of the Supervisory Committee:
Assistant Professor Dong Si
Computing and Software Systems

Many individuals with mental health issues cannot get access to professional help due to reasons such as lack of awareness, limited availability, and high costs. Conversational agents present a viable alternative to deliver mental health support that is accessible, affordable, and scalable. However, the effectiveness of these agents can vary among users, as different users have different personality types such as extroversion, agreeability, etc. which influence how users interact with chatbots. Therefore, it is important to develop therapy chatbots that adapt to individual personalities. In this study, we highlight the significant role of Personality Adaptive Conversational Agents (PACAs) in mental healthcare. We designed an architecture around traditional ML models and open-source LLMs to build a PACA for mental health (based on the existing iCare project at the DAIS research group at UWB). We utilized the architecture to build a functional prototype and conducted a user study, which concluded that personality adaptiveness is a critical feature for mental health chatbots. The prototype is currently live and freely available for use at <http://test.icare.uw.edu:3010/>.

TABLE OF CONTENTS

List of Figures	iii
List of Tables	vi
Chapter 1. Introduction	8
1.1 Motivation.....	8
1.2 Problem Statement	8
1.3 The iCare project.....	9
1.4 Adaptive CAI in Mental Health	11
Chapter 2. Literature review	12
Chapter 3. Methodology	15
3.1 Personality Profiler	18
3.1.1 Personality theories and datasets	19
3.1.2 Resampling approaches	21
3.1.3 Baseline Logistic Regression Model	21
3.1.4 BERT base	22
3.1.5 DistilBERT model	23
3.2 Context-building using LLMs.....	24
3.2.1 Orchestration frameworks.....	26
3.2.2 Prompt Engineering	27
3.2.3 Hyperparameter Tuning	28
3.2.4 Fine-tuned response generation	29

3.2.5	Optimization methods	31
3.3	Summarization Pipeline	32
3.3.1	Fine-tuned summarizer	33
3.3.2	Prompted summarizer	35
3.4	Setup and Deployment	35
3.4.1	Frontend	36
3.4.2	Backend.....	36
3.4.3	Microservices	37
3.5	User Study.....	38
3.5.1	IRB certification.....	38
3.5.2	Online survey	39
Chapter 4.	Results	39
4.1	Personality Models.....	40
4.2	Open-source LLMs	43
4.3	Response Generation	49
4.4	Survey Results	53
Chapter 5.	Conclusion.....	57
5.1	Future plans.....	58
Bibliography	60
Appendix A	63
Appendix B	65

LIST OF FIGURES

Figure 1-1 Types of mental health services received among adults aged 18 or older in the year 2021 (Source: Substance Abuse and Mental Health Services Administration [1])	9
Figure 1-2 A sample of a conversation while interacting with Carebot (top) and Coachbot (bottom). The responses shown are from version 1 of the iCare application, where the project stood before this research.....	10
Figure 3-1 The overall architecture diagram to enable the personality-adaptive feature in the Carebot. The flow of data between the frontend, the backend, and the ML services starting with the <i>User Input</i> and ending at the <i>Carebot response</i> makes up the PACA portion.	17
Figure 3-2 PACA breakdown for a single message-to-response scenario.....	18
Figure 3-3 Data flow diagram for a sequence of messages exchanged between the PACA and a sample user (The personality types are based on the Big Five Personality theory)..	18
Figure 3-4 The step-by-step breakdown of the working of the personality profiler and how it makes use of the user, personality model, and database. Currently, the average of the personality scores is taken into consideration while inferring the user’s personality	19
Figure 3-5 Personality counts of the non-OCEAN data frame before and after oversampling	21
Figure 3-6 Plots showing training loss, training accuracy, validation loss, and validation accuracy for the fine-tuned BERT model on a normal data frame (left) and oversampled data frame (right)	23
Figure 3-7 Plots showing training and validation accuracies (left), and training and validation losses (right) for the fine-tuned DistilBERT model on the resampled dataset	24
Figure 3-8 Information flow diagram between the user and CAI with and without using context. Context-rich setup has a bidirectional flow of information between CAI and Conversation history while context-free setup has a unidirectional flow of information	25
Figure 3-9 Breakdown of the Carebot service as a PACA based on the personality scores and the conversation history stored in the user’s existing profile in the database.....	27

Figure 3-10 Training loss values at every 25 steps of the training epoch while fine-tuning the Llama2 7B chat model on the Counsel dataset	31
Figure 3-11 Data flow diagram in the process of summary report generation	33
Figure 3-12 Django backend setup of the iCare website (figure contributed by Joe Berria)37	
Figure 4-1 Confusion matrix for true labels and predicted labels given by the BERT base model (left) and the DistilBERT model (right) on the resampled data frame	43
Figure 4-2 Sample conversation between a mental health patient and chatbot using Mixtral AI	43
Figure 4-3 Sample conversation between a mental health patient and chatbot using Llama2	44
Figure 4-4 An excerpt of the conversation between a possible iCare user and personality-adaptive Carebot (the test user was given the ‘agreeability’ type as the highest personality score).....	46
Figure 4-5 Difference in the responses generated by the LLM for a typical user and a neurodivergent user (every other parameter kept the same except the user type in the system prompt).....	47
Figure 4-6 Personality scores generated by the personality detector in the backend for a sample incoming message from the user.....	48
Figure 4-7 Line chart showing the average response time per token in milliseconds for top_p and temperature values each ranging between 0 and 2. p controls the cumulative probability of the tokens considered for next token generation. temp controls the probability	50
Figure 4-8 Plot showing the average response time per token in milliseconds for different top_k values. k controls the number of tokens considered in the token generation process50	
Figure 4-9 Average time taken by each response given by the model while using different optimization methods (the lower, the better)	52
Figure 4-10 Number of tokens generated in each response given by the model while using different optimization methods (the lower, the better for our use-case).....	52
Figure 4-11 Identification of the important features that any mental health therapy chatbot should incorporate	54

Figure 4-12 User responses when reporting if they agree with a particular statement regarding their interaction with both versions of the Carebot..... 55

Figure 4-13 User responses when reporting if they strongly agree with a particular statement regarding their interaction with both versions of the Carebot 56

Figure 4-14 User responses answering yes or no to different affirmative questions regarding their interaction with both versions of the Carebot 57

LIST OF TABLES

Table 1: Accuracy values for the Logistic regression model, the BERT base model, and the DistilBERT model before and after resampling	40
Table 2: Personality scores given by the DistilBERT model for a sample test dataset. The scores in bold show that the highest-scored personality types matched the actual personality type of the text in question	40
Table 3: F1-score achieved by the BERT base model with and without oversampling for all the personality types. The scores also include the number of supporting data (in brackets) in testing the selected personality type	41
Table 4: F1-score achieved by the DistilBERT model with and without oversampling for all the personality types. The scores also include the number of supporting data (in brackets) in testing the selected personality type	42

ACKNOWLEDGEMENTS

I'm highly indebted to the constant support and guidance from my Capstone Committee Chair Dr. Dong Si and my capstone committee comprising Dr. Annuska Zolyomi and Dr. Muhammad Aurangzeb Ahmad. I'm also thankful to all the members of the DAIS research group and especially, the iCare project founders Raghav Naswa and Remya Mavila, and the undergraduate Capstone students Joe Berria and Joyce Lee for the successful completion of my thesis.

Chapter 1. INTRODUCTION

1.1 MOTIVATION

Some of the major reasons for the current poor status of mental health services are the lack of awareness about the seriousness of mental health problems among the majority of the population, and the limited access to mental health facilities for those who are aware of mental health problems. Additionally, the limited mental health manpower involved in providing psychotherapies and counseling makes the cost of therapy sessions unaffordable for the common man. Thus, to create easily accessible and affordable assistance for mental health care, the utilization of Conversational AIs (CAIs) is an opportunity worth pursuing. CAIs are artificially intelligent agents that can engage in contextually rich conversations with users, and they are broadly used to refer to chatbots, virtual agents, etc. They can attempt to provide many benefits that a psychotherapist could offer, such as engaging the user in deep conversations, building relationships with the user, and providing apt responses, while avoiding some demerits like patient disclosure (if the platform serving the CAIs has a good mechanism for encryption and storage of user's data) and preconceived bias (depending upon the training data used).

1.2 PROBLEM STATEMENT

The number of adults turning to virtual services for mental healthcare is quite high, third only to combined mental health services and prescription medication (Figure 1-1). However, there are only a handful of such applications or services available today that could provide reliable mental health therapy. The small proportion of these services that do a good job of providing therapeutic responses currently do not have a mechanism to personalize the responses to the needs of the

individual patient due to which they are more likely to give similar recommendations to each patient suffering with a particular issue even though the actual recommendation required might differ from person to person. Hence, it is important to build adaptive and personalized virtual services to possibly elevate the impact of virtual services in mental health therapy.

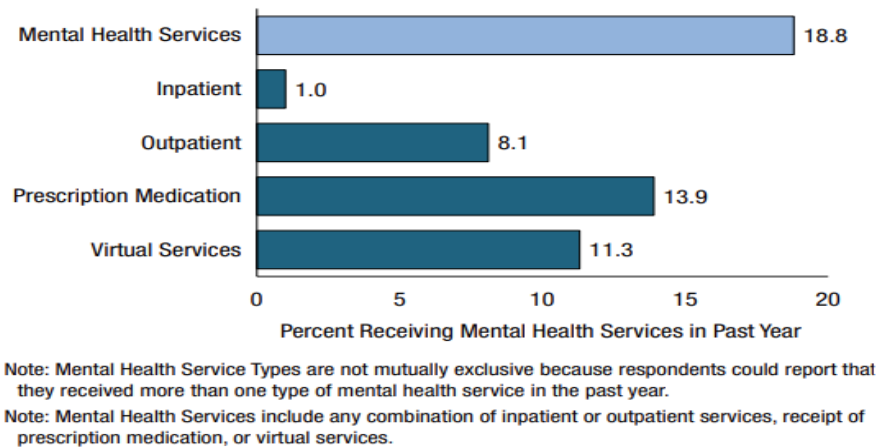


Figure 1-1 Types of mental health services received among adults aged 18 or older in the year 2021 (Source: Substance Abuse and Mental Health Services Administration [1])

1.3 THE ICARE PROJECT

This research was based on the iCare Project of the Data Analysis and Intelligent Systems (DAIS) research group at the University of Washington, Bothell, and the associated development was meant to minimize the limitations of the project. The iCare ecosystem currently has two mental health chatbots - Carebot and Coachbot. The Carebot provides empathetic responses to users suffering from mental health issues. At the same time, the Coachbot engages the users, who are taking care of other mental health patients, in conversation to improve their caregiving traits.

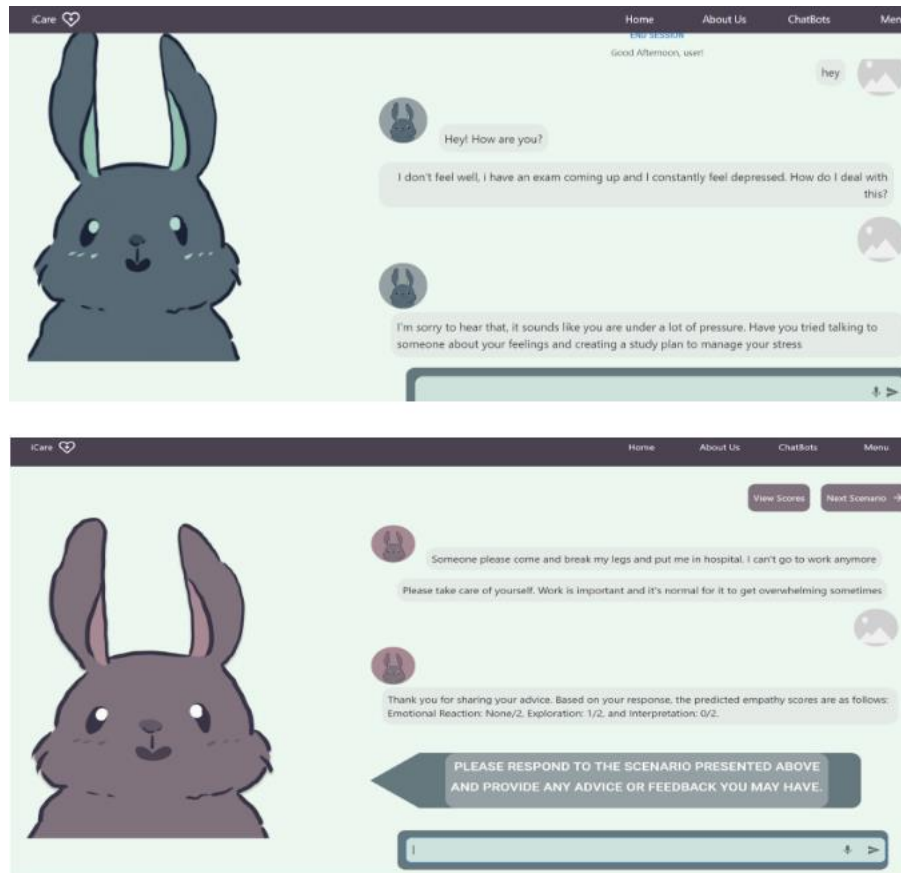


Figure 1-2 A sample of a conversation while interacting with Carebot (top) and Coachbot (bottom). The responses shown are from version 1 of the iCare application, where the project stood before this research

Before this research, the Carebot identified the intent of the user's input and generated responses using the RASA framework with GPT3 as a fallback model. It could understand most mental health issues the user might be experiencing and provide a well-suited empathetic response as shown in Figure 1-2. The Coachbot scored the empathy level of the caregivers based on their response to a particular mental health scenario (as depicted in Figure 1-2) by using 3 different large language models (LLMs) fine-tuned on mental health datasets based on the EPITOME framework. The abilities of both these chatbots were inhibited by the fact that they did not learn from the users' inputs over time and none of the generated responses had any relationship to one another, which

could heavily compromise the user experience and would not make the chatbot the preferred tool for improving mental health outcomes. Hence, making the chatbots adaptive was a promising way to build a foundation for a strong mental health environment.

We confined our research to the study and improvement of the Carebot functionalities. Our research efforts included refactoring the complete iCare structure to fit a personality-adaptive conversational agent (PACA) into the Carebot. PACAs are artificially intelligent agents that adapt to the user's personality on their own by inferring the personality type of the user from the user's messages and comments over time.

1.4 ADAPTIVE CAI IN MENTAL HEALTH

Natural Language Processing (NLP) and Machine Learning (ML) are used by CAIs to engage with users in human language in the form of voice assistants, virtual assistants, or chatbots. They can assist the intended users with various functions such as information retrieval, customer service, entertainment, and healthcare.

Adaptive CAIs are the classes of AI systems that can adapt and improve as they encounter changes in input data and the use case environment, unlike traditional CAIs that follow set algorithms and decisions. If the adaptive CAIs can provide empathetic responses, users are more likely to share their feelings with them since the users would also be receiving responses that suit them instead of some general responses. Thus, by offering individualized and sympathetic support, adaptive CAIs can assist users in coping with mental health concerns.

The central idea of our research was to explore the use of adaptive CAIs in the context of mental health services and examine the performance of an adaptive mental health chatbot by actually building it. The primary factor for adaptation we considered in this work was the personality of the user; however, there are many other critical factors that mental health chatbots can adapt to such as specific health issues, behavioral patterns, etc. We focused on personality which is an important trait in healthcare since it can influence how engaged, trusted, and satisfied a user is with the chatbot [2]. Adaptive CAIs can be built to adapt to the user's personality on their own by inferring the personality type of the user from the user's inputs, and by changing their responses as per the personality profile of the user. As a part of this study, the conversational agents could generate and adjust their responses according to the detected personality of the mental health patient using the iCare platform.

Chapter 2. LITERATURE REVIEW

Mental health issues continue to be one of the most widespread yet severely overlooked forms of health issues in the modern world. As per the World Mental Health Report issued by the World Health Organization (WHO) in 2022, mental disorders are the leading causes of “years lived with disability” and are ranked second in the total global years lived with disabilities [3]. Post-pandemic, the prevalence of mental health issues has accelerated even more especially among adolescents and young people with increased depression, anxiety, and psychological distress [4]. Despite the pervasiveness of the problem and the fact that conversation is the most effective method to treat any form of mental health issues [5], a study has shown that more than half of

Americans who need therapy are not able to get it due to lengthy waiting periods, the shortage of practicing therapists, and stigmatization around the topic [6]. Thus, with the rapid progress in conversational AI agents (CAIs) such as ChatGPT, now is the perfect time to leverage technology for the mental healthcare industry [7]. CAIs can influence the domain of mental health by existing in many forms like as therapist, co-therapist, therapist assistant, or therapist trainer [8] and they have the potential to revolutionize mental health care by providing personalized, accessible in terms of availability, and cost-effective support to patients and caregivers [9].

As far as the present state of mental health CAIs or chatbots is concerned, most of them work towards getting people in need of assistance in touch with people who can assist [10]. While their idea is noteworthy, the mental healthcare industry still severely suffers from the typical problems highlighted in [6] like structural and financial barriers. The advanced forms of mental health chatbots like Woebot, Wysa, etc. which utilize ML methods to provide therapeutic responses are landmark achievements in the field of mental health chatbots but their inability to dynamically adapt their responses based on who they are interacting with [7] restricts their potential to make an impact on a larger scale. Since psychotherapy is highly patient-centered in clinical practice, analyzing the user's data and providing personalized mental care seems an effective approach [11], giving rise to the need for adaptive CAIs.

Adapting to the user's personality is the most important criterion for a chatbot in mental health, as the success of such chatbots depends on user trust [12]. Since each user has a unique personality which can be inferred from the trends in conversational history, the chatbots can learn from these and use the appropriate words in the responses hence creating PACAs. Previous research has

shown that personality matching can enhance the user experience and satisfaction with chatbots [13] and it has been quantitatively proven by the authors of [2] that mental health chatbot personality has a positive impact on user engagement since the session duration of the conversation and number of messages exchanged increase if the personalities match. PACAs should use a reliable and valid personality model to evaluate the user's personality traits, regardless of the decision-based, ML-based, or hybrid approach, although many personality models can be used to detect the personality of the user [14]. One of the most important such models is the Big 5 personality model which classifies personality in terms of five basic dimensions – agreeableness, neuroticism, extraversion, openness, and conscientiousness [15]. Among all the personality dimensions, agreeableness and extraversion are the most important personality dimensions in terms of interpersonal interactions [16]. Developers have also built a chatbot to detect and analyze the user's emotions based on their text inputs and provide personalized responses using a lexicon-based approach and an emotion ontology, which is a promising way to build PACAs using cognitive skills [17]. However, there are also ethical concerns associated with these chatbots. Confidentiality and data privacy are significant areas of concern with other added concerns like lack of proper standardization, misdiagnosis, and inappropriate responses [10], [18]. Our research considered all the valid concerns and proposed the architecture design to build a PACA on top of already existing mental health chatbots. With this research, we aimed to strengthen the existing state of online psychotherapy by running a personality detector in the backend of mental health chatbots and feeding that into Generative AI to give responses while maintaining as much security as possible.

Chapter 3. METHODOLOGY

We designed the PACA architecture so that it could adapt the responses according to the personality profile of the user created over time. The backend was planned to store the modifiable personality profile of each user. Whenever there was a new conversation from the user's end, the next response was adjusted according to the combination of the personality identified in the incoming message and the past personality profile formed for the user. The personality-adaptive nature of the system was based on how the personality model and the LLM came together to create a pipeline between when the user entered a new message and when they got the response back from the Carebot as depicted in Figure 3-1. The overall architecture was also accompanied by the summarizer module which was meant to better understand the user's personality over past sessions.

We constructed the technical design of the PACA in a way that both the personality model and the LLM got activated for every single message-to-response flow as shown in Figure 3-2. Each incoming message first underwent the personality model. The personality model was placed so that it could extract the past personality profile of the user from the database while calculating the personality scores for the new message. It updated the personality profile in the database with the new scores via averaging and passed the incoming message along with the updated personality profile onto the LLM. The LLM was positioned so that it could fetch the older conversation of the user from the database, and make use of the past conversation, the new message, and the personality profile to create the adapted response. In the end, the LLM would update the conversation in the database with the newly created response and send the response back to the user, hence completing the message-to-response pipeline for a single conversation exchange. For

each session, the same pipeline occurred in a loop for every new message and its response as broken down in Figure 3-3. For the situations where the PACA was not able to accurately estimate the personality type of the given user, we configured the setup in a way that the user would be associated with the 'neutral' personality type. This entire setup allowed the agent to have a modifiable personality profile for each user and also calculate the personality by making use of all the past user messages including the latest one.

Section 3.1 breaks down the efforts that went into building the personality classification models, section 3.2 gives an overview of how to build context to convert a CAI into a PACA, section 3.3 describes the goals and steps to create the summarization pipeline, section 3.4 goes through the full-stack setup of the restructured iCare project, and section 3.5 introduces the nature and the process of user study conducted to study the impact of our PACA in mental healthcare.

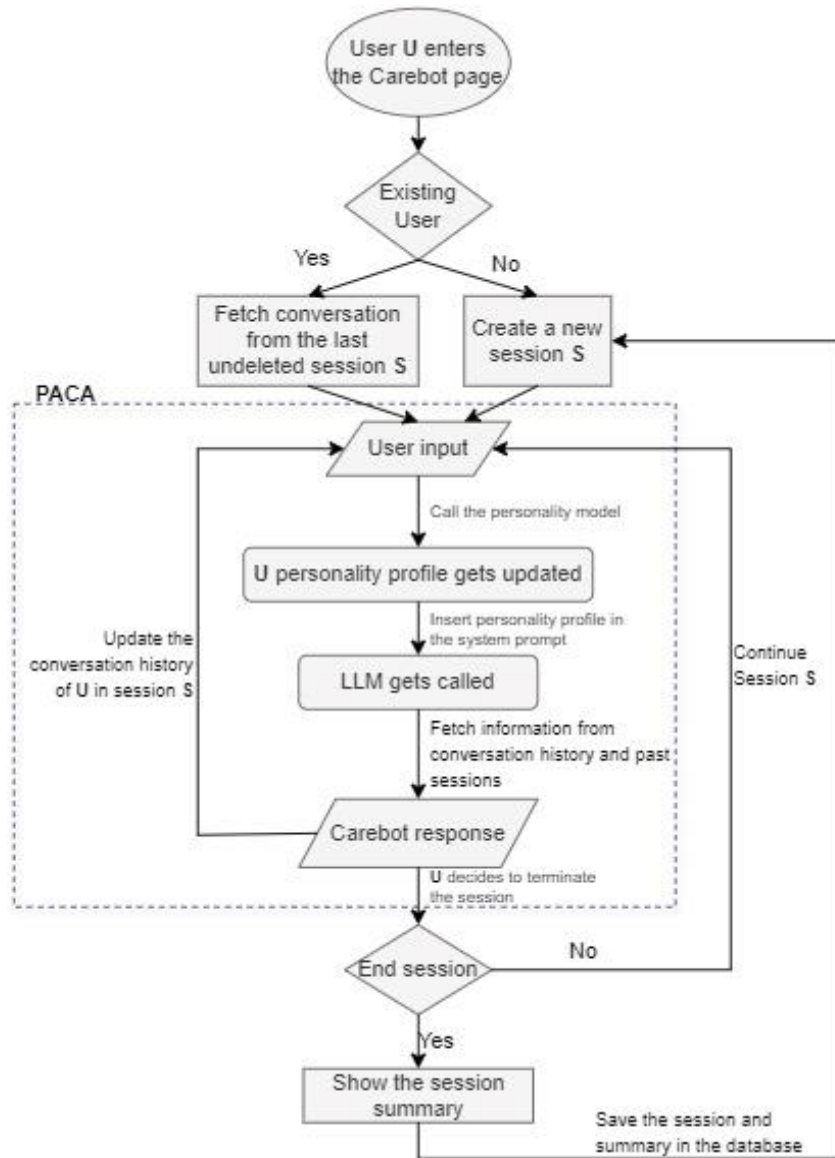


Figure 3-1 The overall architecture diagram to enable the personality-adaptive feature in the Carebot. The flow of data between the frontend, the backend, and the ML services starting with the *User Input* and ending at the *Carebot response* makes up the PACA portion.

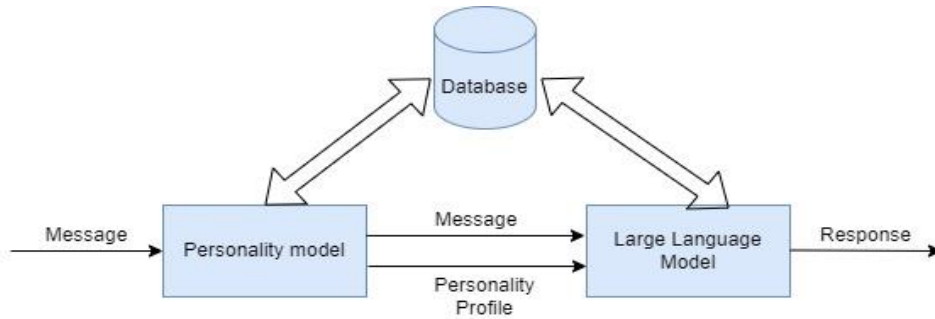
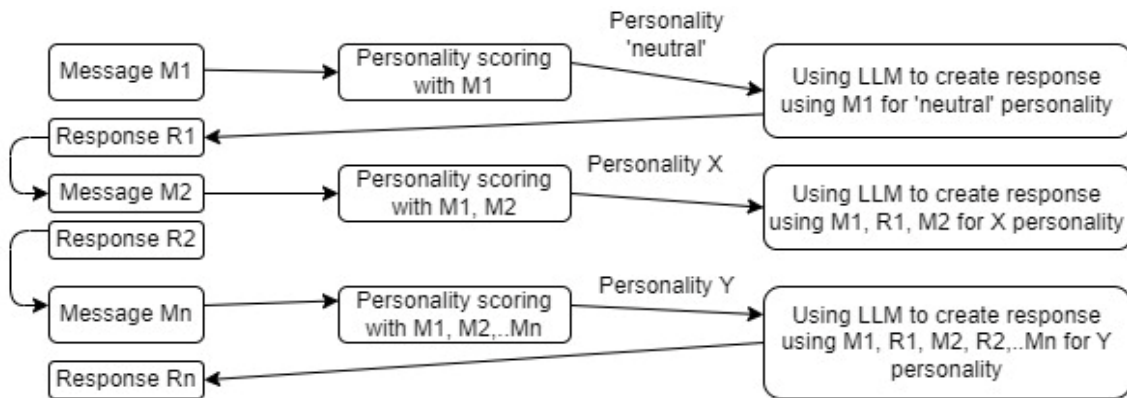


Figure 3-2 PACA breakdown for a single message-to-response scenario



X, Y = extraversion | agreeableness | neuroticism | openness | conscientiousness

Figure 3-3 Data flow diagram for a sequence of messages exchanged between the PACA and a sample user (The personality types are based on the Big Five Personality theory)

3.1 PERSONALITY PROFILER

The adaptive ability of the chatbot was based on the system’s ability to create a different personality profile for different users of the platform. We achieved this by building and integrating a personality profiler after the chatbot receives new user input and before the input gets fed into the underlying LLM to infer the responses. A personality profiler was designed and developed based on NLP models and trained for personality classification using the Big Five personality theory. We refer to these models as “personality models” in our study. As shown in Figure 3-4,

the personality model gave the probability values for each new message as per the personality dimensions in the Big Five Personality theory. The system maintained the personality profile history of each user in the database. As soon as the new values (or scores) were generated for a particular user's message, the personality profiler worked to add those new scores to the existing set of scores in order to calculate the average of all the scores, hence updating the personality profile for the user.

3.1.1 *Personality theories and datasets*

We considered some of the most popular personality theories to develop and fine-tune the personality models and chose the Big Five personality theory for its comprehensiveness and wide applicability across various contexts. The Big-Five personality Theory divides the dimensions into 5 dimensions: Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism, and the theory itself has been extensively validated through cross-cultural studies [15].

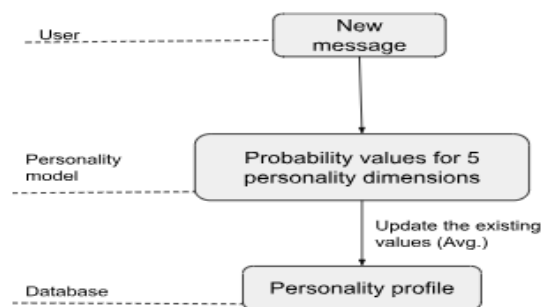


Figure 3-4 The step-by-step breakdown of the working of the personality profiler and how it makes use of the user, personality model, and database. Currently, the average of the personality scores is taken into consideration while inferring the user's personality

There also exists a more advanced personality model called Cattell's 16 Personality Factor Model which categorizes the personality dimensions into 16 factors including emotional stability, warmth, liveliness, etc. [19] but was discarded due to its complexity and some overlaps with the

Big Five dimensions. We also considered the Myers-Briggs Type Indicator (MBTI) which uses four dichotomies for personality: extraversion-introversion, thinking-feeling, sensing-intuition, and judging-perceiving [20]. Though MBTI has more popular appeal and is easy to understand, we chose the Big Five dimensions over MBTI since MBTI divides the user into types while the Big Five uses dimensions that give a more nuanced understanding of the user's personality, which suited our research use-case better.

The referenced dataset gave five personality scores to 307,313 people from different countries, which was calculated using the guidelines from the International Personality Item Pool (IPIP). The dataset originally consisted of broken-down datasets where one dataset stored the user input for different case IDs while the other recorded personality scores for the users. The CSV datasets were converted into respective data frames using the Pandas library and then merged on their case ID values to arrive at a combined data frame.

A new personality column was also created in the final data frame to be used as the dependent variable of our models which put the case IDs into the personality class in which the case ID had the highest score. The personality column was further transformed into quantitative from categorical to ease the processing. Hence, the final data frame ended up having labels ranging from 0 to 4. We also used the OCEAN dataset [21] of personality to build our models, as the OCEAN dataset had more consistent data items for our use case. Both datasets underwent the same processing steps to arrive at the final training and test sets.

3.1.2 Resampling approaches

Resampling is a statistical method used in ML to generate new data points in a dataset by randomly picking data points from the existing dataset. It helps in creating new synthetic datasets for training ML models and in estimating the properties of a dataset when the dataset is unknown, difficult to estimate, or when the sample size of the dataset is small.

We implemented oversampling methods to get rid of the imbalanced nature of the dataset. After trying a variety of oversampling methods like SMOTE-Tomek, Random Over sampler, and Custom Oversampling, we concluded that Custom Oversampling resulted in the most uniform distribution of data and gave the best results for our baseline models. To arrive at the custom oversampled dataset as in Figure 3-5, duplication and synthetic modification of underrepresented personality items were followed. The texts with agreeable traits were multiplied twice, the extraversion data items were duplicated thrice, and so on.

The diagram illustrates the process of oversampling. It shows two dataframes: 'Original dataframe' on the left and 'Resampled dataframe' on the right. An arrow labeled 'Oversampling' points from the original to the resampled dataframe. The original dataframe has counts for Agreeable (55), Extraversion (44), Openness (87), Conscientiousness (63), and Neuroticism (25). The resampled dataframe has counts for Agreeable (110), Extraversion (132), Openness (87), Conscientiousness (125), and Neuroticism (100).

Personality	Count
Agreeable	55
Extraversion	44
Openness	87
Conscientiousness	63
Neuroticism	25

Original dataframe

Oversampling

Personality	Count
Agreeable	110
Extraversion	132
Openness	87
Conscientiousness	125
Neuroticism	100

Resampled dataframe

Figure 3-5 Personality counts of the non-OCEAN data frame before and after oversampling

3.1.3 Baseline Logistic Regression Model

Logistic regression is a statistical method used for classification problems. In text classification, logistic regression is used to predict the probability of a given text belonging to a particular class.

CountVectorizer is a tool in Python's Scikit-learn library that is used to convert a collection of text documents into a matrix of token counts. During prediction, the same CountVectorizer is used to convert the new text into a matrix of token counts, which is then fed into the trained logistic regression model to predict the probability of the input belonging to a particular class.

To create the baseline model, 70% of the processed data was used for training and 30% for testing. The Count Vectorizer was prepared using the texts in the training dataset and then used to transform both training and test sets. The same method was followed for both unsampled and resampled data.

3.1.4 *BERT base*

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained deep learning model that can be fine-tuned for various NLP tasks, including text classification [22]. BERT models are usually pre-trained on a large corpus of text and then fine-tuned for specific tasks. The BERT family of models is available on Hugging Face. These preprocessing models tokenize the input text and inputs are then fed into the BERT encoder to compute vector-space representations of text. The resulting vectors can be used for text classification.

The BERT model was trained using the BERT preprocessors and encoders available in the Tensorflow Hub. 5 neurons based on the soft-max activation function were added on top of the pre-trained model to match the required output. Categorical cross-entropy was used as the loss function and early stopping was used during training to track the validation loss. The final model had a significantly different pattern of training for the original data and oversampled data. The BERT model fine-tuned on the original dataset followed a typical pattern of losses where

validation accuracy significantly picked up over the training while both training and validation losses saw dips as seen in Figure 3-6.

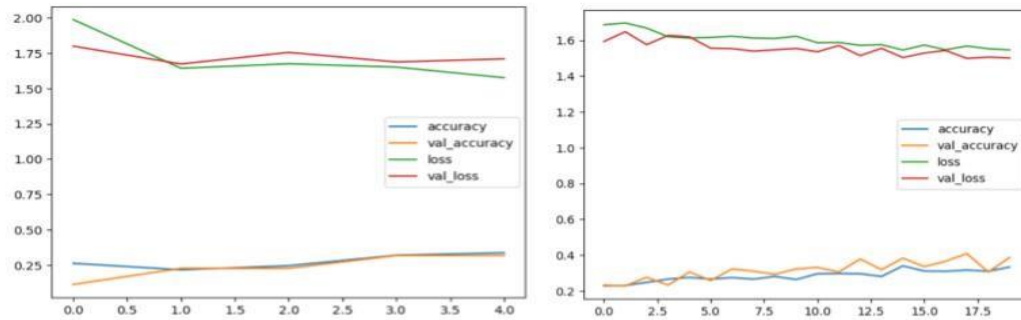


Figure 3-6 Plots showing training loss, training accuracy, validation loss, and validation accuracy for the fine-tuned BERT model on a normal data frame (left) and oversampled data frame (right)

On the other hand, when the model was trained on the oversampled data, both validation accuracy and validation loss followed a zigzag pattern throughout the training, implying that the model could not fit the data during training.

3.1.5 *DistilBERT model*

The DistilBERT model is one of the most powerful models for text classification pre-trained on a large corpus of text data using a masked language modeling objective and learns contextualized word representations by considering both left and right context [23]. The key feature of this model is “knowledge distillation” which enables the model to mimic the behavior of the larger model (like BERT) but with fewer parameters. The training scheme used for the DistilBERT model was kept the same as the other models for consistency. As shown in Figure 3-7, the losses decreased throughout the training for both training and validation sets, which remarkably increased the accuracy for both.

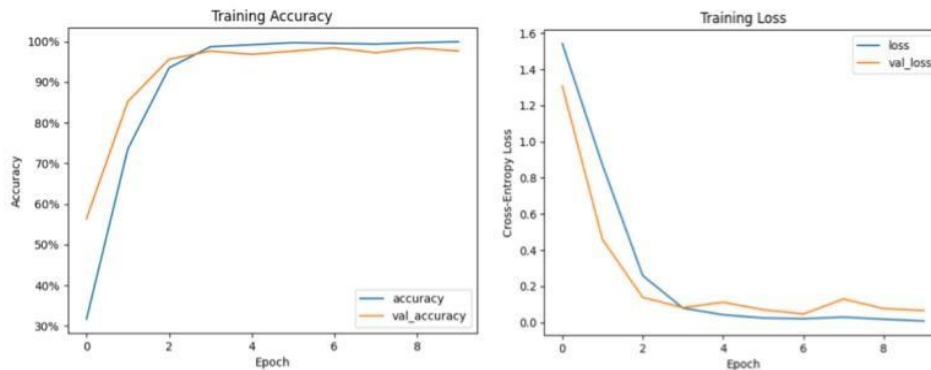


Figure 3-7 Plots showing training and validation accuracies (left), and training and validation losses (right) for the fine-tuned DistilBERT model on the resampled dataset

3.2 CONTEXT-BUILDING USING LLMs

LLMs are sophisticated AI systems that can understand and generate human-like text by learning from vast amounts of existing written content. The LLMs that have publicly accessible model architectures, source code, and weight parameters are called open-source LLMs. They are highly customizable in terms of inspection, evaluation, etc. which also increases the possibility of malicious use. On the other hand, the LLMs whose model architecture and weights are proprietary assets are termed closed-source LLMs. They excel at almost all natural language tasks together with being the best preference as far as security is concerned. They allow less customization, and they have to be mostly accessed via paid API keys which makes the experiments costly.

The central idea of our research was the feature of adaptiveness, which was possible through context-building. Context serves as a memory for a CAI that helps the CAIs remember what happened earlier in the conversation/session. It helps the chatbots keep track of conversation history and pick up from where the conversation was left off. Context-rich agents have read/write

access to the information stored in the conversation history, unlike context-less agents which only have write access (as shown in Figure 3-8). In other words, the context-less agents are only able to pass information on to the conversation history while the context-rich agents can maintain back-and-forth communication with the conversation history.

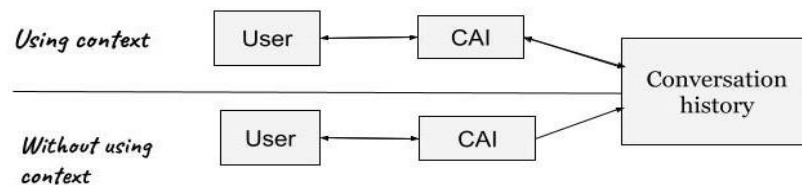


Figure 3-8 Information flow diagram between the user and CAI with and without using context. Context-rich setup has a bidirectional flow of information between CAI and Conversation history while context-free setup has a unidirectional flow of information

Alongside the actual conversation, the conversation history can be designed to hold vital information about the users to learn more about them, personality type being one such information. Context-building was the mechanism that transformed the typical CAI into a PACA. Upon each new input from a user, the PACA could pull the personality profile of the user from their conversation history and use the information as the context for new responses. Using context enabled the PACA to move away from general responses and create responses based on the personality profile. For instance, if the personality profile of the user indicated the probability of a user being an introvert was more than the probability of them being agreeable, the responses given by the PACA would make use of this information while creating the responses. Many powerful libraries allow NLP developers to build context-rich chatbots. For our experiments, we explored different LLM orchestration frameworks. However, the process of setting up the context could also be achieved without the established libraries by just using the conversation history (stored in

accepted formats like text, JSON, Python dictionaries, etc.) and relying entirely on prompt engineering as needed.

3.2.1 *Orchestration frameworks*

Orchestration frameworks are tools and libraries that manage the coordination and interaction between multiple small and large language models to ensure seamless setup and usage of context-aware responses.

One of the most used LLM orchestration frameworks is LangChain. It is an open-source framework designed to simplify the creation of applications using LLMs. It offers tools for building personalized, efficient, and engaging chatbots. Conversational chains built using LangChain enable chatbots to track the conversation history and relevant context, resulting in more natural and personalized conversations. It provides the flexibility of using a large variety of LLMs ranging from open-source ones to closed-source ones.

Another promising library is Ollama which is a growing open-source framework that provides a lightweight and ready-to-use solution for running and customizing LLMs locally without depending on paid commercial APIs. It provides a list of the most up-and-coming open-source LLMs to use in a local environment and it streamlines the integration process by packaging the model weights and configuration into a single Modelfile. The framework allows customization of pre-built models using prompts and system messages which helps the LLMs achieve the desired context for their responses.

3.2.2 Prompt Engineering

Prompt engineering is the process of refining input prompts to guide Generative AI models in creating specific and high-quality outputs. It ensures that the underlying LLM understands the language and the intent behind the user queries. In our use case, prompt engineering did not only guarantee that the LLM behaved as a therapist but also made sure the LLM made use of the personality profile of the user of the Carebot service. As shown in Figure 3-9, this was achieved by inserting the system prompt section of the Carebot as a bridge between the actual LLM and the user personality profile stored in the database.

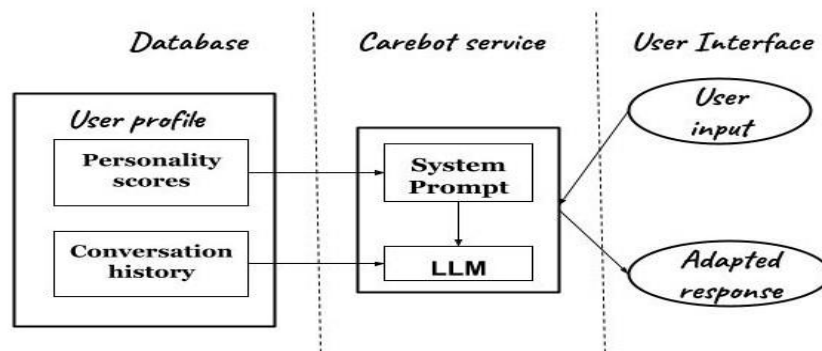


Figure 3-9 Breakdown of the Carebot service as a PACA based on the personality scores and the conversation history stored in the user's existing profile in the database

The backend architecture of the PACA was divided into the Carebot service and the database. While the service was responsible for taking in the new message and providing the adapted responses, it did so by making use of the information stored in the database. Within the Carebot service, the system prompt utilized the personality scores to learn the possible personality traits of the user. The prompt was then passed on to the chosen LLM (i.e. Llama2), which already had access to the previous conversation history. The LLM took in the user input along with the prompt and the conversation history and gave the personality-adapted responses. As the conversation

progressed, the prompt updated itself with each change in the personality scores while the LLM maintained the connection with the updated conversation history. As a result, the setup allowed the Carebot to dynamically update its responses according to the personality of the user, giving rise to the personality-adaptive nature of the PACA.

A lot of the state-of-the-art LLMs including the Llama2 7B chat model has an option to use the system prompt in creating conversational applications. These are the prompts, different from the input prompts from the user, which dictate the behavior of the model as long as the conversation session continues. We explored the complete architecture with a lot of different system prompts to arrive at a final prompt giving consistent and acceptable responses. While we created multiple different system prompts as a part of this study, the common characteristics shared by all of them were that i) they had to make use of the personality scores from the backend, ii) they needed to act as a mental health therapist, and iii) they had the motive to engage the user in a conversation.

3.2.3 *Hyperparameter Tuning*

In addition to the prompt used, the other crucial factors that control the text generation of an LLM are the hyperparameters associated with the LLMs [24]. We experimented with 3 major hyperparameters viz. Temperature, Top-k, and Top-p to manipulate and identify the best set of values for the optimal model output. The text generation function available in the Transformers library was used to play around with different combinations of these hyperparameters to arrive at the conclusive set of parameters. We ran the experiments setting different ranges of values for these 3 hyperparameters keeping the same test environment and the same set of simulated user messages. The average number of tokens generated per second and the average time taken per

generated token were studied at different values to tune the parameters into the best possible combination to minimize the response time and maximize the quality of the tokens generated.

The sampling strategy of the model was first set to “True” to allow the usage of hyperparameters, otherwise, the model would just follow the default sampling i.e. Greedy Sampling, which works by prioritizing the highest probability token to create the responses. Temperature is the hyperparameter that controls the randomness of the generated output [25]. A higher temperature makes a bigger curve representing the set of next tokens leading to the responses being more diverse and creative. The default value of Temperature is set to 1 in the text-generation pipeline while we set the range from 0 to 2 to generate the responses. The Top-k value limits the selection of tokens to the top-k most probable ones leading to the smaller values generating more focused responses and the larger values giving varied responses [25]. We set the Top-k values ranging from 0 to 20 while the default is set at 50. The Top-p (also known as Nucleus Sampling) value restricts the cumulative probability of tokens generated during sampling and selects the most probable tokens whose cumulative probabilities add up to at least p [25]. Controlling the top-p value helps to create coherence and creativity in the responses. The default Top-p value set in the Transformers library is 1 and we ran the response generator for different Top-p values between 0 and 1.

3.2.4 *Fine-tuned response generation*

Fine-tuning options were also explored and studied as a part of our research to improve the quality of the responses generated by the Carebot. Though the base model gave appropriate and within-context responses most of the time, there were instances when the model would divert to giving generic responses if the topic being discussed in the session was more general in nature. We, however, wanted to avoid that as it would not be an ideal application of a mental health chatbot to

discuss generic topics and explored the benefits of fine-tuning process to generate the responses. We mainly stuck with different fine-tuned instances of the Llama2 [26] and Llama3 [27] models for making the comparative analysis and reported the major performance differences between the base model and the fine-tuned models. In the process, we discovered many promising datasets and models already fine-tuned and publicly available for usage. The most popular dataset and the one that was most downloaded on the Hugging Face platform was found to be the Counsel Chat dataset which consists of data from individuals seeking assistance from licensed therapists and their associated responses. Though there are more refined synthetic datasets out there for mental health counseling conversations, we proceeded with fine-tuning the model on the real-world dataset i.e. Counsel Chat dataset [28]. Since there was not a promising fine-tuned instance of the Llama2 model on the mentioned dataset (at the time of this research), we created a fine-tuning pipeline for the same.

Parameter-Efficient Fine-Tuning (PEFT) [29] mechanism was used to minimize resource usage during training and maximize the performance of the trained model. PEFT helps to retrain only some of the weights of the underlying model while freezing most of the weights. Before training, we converted the dataset into rows of Llama2-supported prompt templates. While training, we used Quantized Low-Rank Adaptation (QLoRA) to fine-tune the model in 4-bit precision and reduce the Virtual RAM (VRAM) usage in the training environment. The change in the training loss, as the training proceeded, is shown in Figure 3-10. The loss values were logged at every 25 steps of the training.

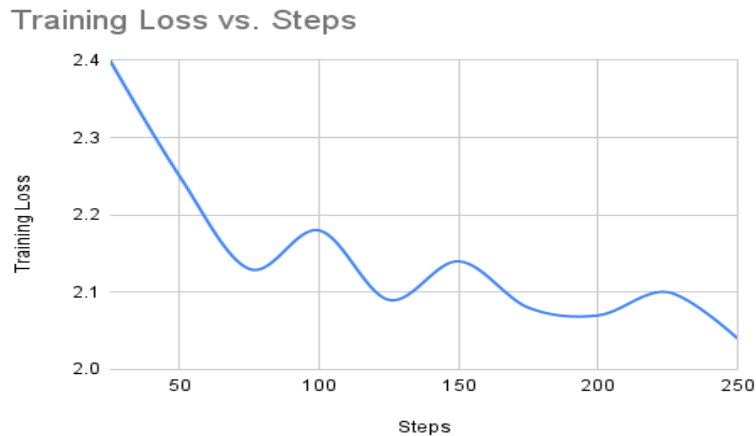


Figure 3-10 Training loss values at every 25 steps of the training epoch while fine-tuning the Llama2 7B chat model on the Counsel dataset

The fine-tuning was performed using the learning rate of $2e-4$ and weight decay of 0.001. We set the LoRA attention dimension rank at 64, the alpha perimeter for LoRA scaling at 16, and the dropout value at 0.1. A batch size of 4 was used per GPU per training.

3.2.5 Optimization methods

We made use of multiple optimization methods and also studied their impact in order to arrive at a setup that would have the most optimal runtime, least response time, least usage of memory, and highest quality of output tokens.

In order to achieve faster inference and easy deployment of the LLM in our test servers, we loaded the model with float16 precision instead of its default precision, which significantly reduced memory usage and increased computation efficiency. For further optimization, we also configured the backend to load the model in 4 bits instead of the original 32 bits. 4-bit quantization of weights reduced the memory required to store the model. This optimization enabled faster calculations,

leading to reduced latency and increased throughput. In order to optimize the fine-tuning, we employed quantized low-rank adaptation (discussed in section 3.2.4).

We also experimented with an external framework called vLLM [30] to achieve optimization of the inference time of our setup. vLLM is a framework designed to enhance the efficiency and scalability of deploying language models. The framework helps in faster and more efficient model inference by leveraging a highly optimized virtual memory management system. It is built to accommodate the complexities of model loading, execution, and data transfer, ensuring that memory usage is minimized, and computation resources are maximized.

3.3 SUMMARIZATION PIPELINE

The summarization pipeline got activated whenever the user requested to end the session. When this pipeline executed, the summary of the conversation history was created, saved into the database, and displayed to the user. In the process, the module also called the personality detector on the user side of the conversation and cleared the conversation session. The module took the text inputs from the conversation as an array and generated a report that comprised the summary of the user inputs and recommended prescriptions for the user. In addition to ending the session, the prompted segment of the summarizer was also used to reduce the length of the LLM-generated responses whenever the number of tokens in the generated response crossed a particular threshold.

3.3.1 *Fine-tuned summarizer*

The pipeline feature available in the Python Transformers library was used to form the abstractive summary of the inputs. Figure 3-11 shows the overall process behind the conversion of user inputs into the final summary report. The inputs originally in the form of an array of texts were first concatenated as a single string and then passed onto the pipeline. The pipeline could create a shorter version of a document that captured all the important information. The pipeline was fine-tuned for abstractive summarization using T5, which is a text-to-text transformer pre-trained on a large corpus of text. The summarizer model also made use of different language models available on Hugging Face. The language models were finetuned on a text corpus of lengthy documents and their concise summaries to create summaries of the conversation generated by the Carebot session.

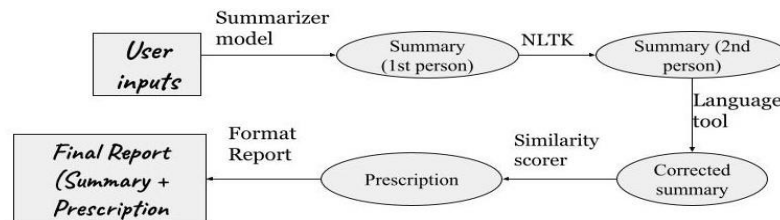


Figure 3-11 Data flow diagram in the process of summary report generation

The summarization pipeline created the summary from the original perspective of the user i.e. the first-person perspective. For the end report, the summary needed to be changed into a second-person perspective for better readability. To do this, Python functions were used to convert all pronouns and possessive pronouns into their respective second-person pronouns. Language tool available in Python was utilized to correct the grammatical errors created during the conversion.

Regular expressions were also used to validate the grammatical correctness of the new summary for further correction and adjustment.

We made use of Spacy to enable the reliable prescription mechanism. Spacy is an open-source library for NLP tasks in Python. It provides a method called similarity that allows us to calculate the semantic similarity of two container objects by comparing their word vectors. It uses Cosine Similarity as the default similarity scorer and the similarity metric ranges from 0 to 1 where a higher score means greater similarity. Another promising option we chose for similarity calculation was the model called Universal Sentence Encoder (USE) which encodes the input text into high-dimensional vectors. USE also made use of cosine similarity to calculate the score between two embeddings and generated better results compared to Spacy.

The prescriptions were generated based on a simple two-column dataset sourced from various mental health references. The first column was the replica of a possible summary of a mental health conversation and the second column was a recommended prescription based on the summary. The similarity scorer did the job of calculating the similarity between the input summary and all the summaries in the dataset. If the score matched the set threshold, the respective content from the prescription column was picked as a possible prescription for the user. In this way, the summarizer concluded its operation by generating a text that was a combination of the generated summary and possible prescriptions.

3.3.2 *Prompted summarizer*

As an alternative way to create a more robust summary, we also used prompt engineering on the Llama2 model [31]. During the end of the session, we used this technique to feed the complete user side of the conversation as a user message to the LLM and created a system prompt to generate a summary that would cover the central topics and problems addressed by the user trimming out the insignificant parts of the conversation. To reduce the length of longer responses generated by the model, a different system prompt was engineered which instructed the model to make the responses concise while maintaining the major information and tone of the response. The inclusion of a summary prompt in the response generator segment helped to maintain longer back-and-forth conversations between the user and the Carebot at the expense of response time.

3.4 SETUP AND DEPLOYMENT

To serve the PACA and make it publicly available in order to understand its possible impact on the mental healthcare industry, it was integrated into the iCare project (as an upgraded version of the Carebot) and deployed using the latest frameworks of web development. Django, Docker, SQL, Fast API, etc. were used to build and serve the backend of the iCare platform while the front end was designed and developed using Next JS, Tailwind CSS, ShadCN UI, etc. The new features were gradually added to the website during development and tested within the University of Washington's Virtual Private Network (VPN). Upon final testing and integration, the prototype was deployed using the public ports available on the DAIS servers. The complete project including the Llama2 model, the supporting models, frontend code, and backend setup is currently deployed

live and makes use of NVIDIA GeForce GTX 1080 * 4 supported by the latest NVIDIA Driver version 535.104.05 and CUDA version 12.2.

3.4.1 *Frontend*

The front end of the iCare website was built using NextJS and styled with Tailwind CSS and ShadcnUI. The UI/UX design was carefully researched for mental health patients and consisted of standard pages like “About”, log in, and Sign-up, alongside a dedicated chatbot interaction page. The website also had a profile page that seamlessly integrated user data and historical conversations retrieved from the backend database. We also focused on security which was achieved through API tokens, ensuring robust communication between frontend and backend components.

3.4.2 *Backend*

The backend structure of the iCare project was built using a popular Python framework called Django, which uses SQLite as the database. In the DAIS servers, the backend was made available on port 8010 while it connected with different AI services on port 5007 through Remote Procedure Call (RPC), as represented in Figure 3-12.

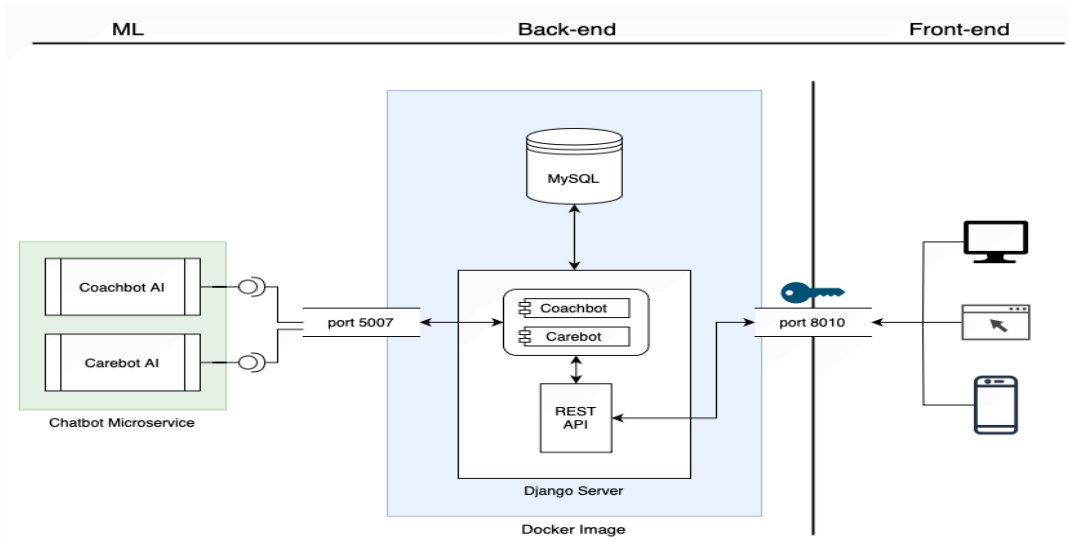


Figure 3-12 Django backend setup of the iCare website (figure contributed by Joe Berria)

It was necessary to decouple the ML features from the rest of the backend functionalities to reduce the disk space needed for the backend Docker image. The decoupling provided better stability for both parts and provided additional security through controlled access.

3.4.3 *Microservices*

The AI services like the Carebot and Coachbot were completely separated from the Django server to seamlessly segregate the ML work of the backend from the non-ML work. The REST API carrying the complete iCare backend was deployed using a Docker image and was developed to not expose the Chatbot microservices to the web. The microservices for the chatbots were designed to run in a separate process from the API gateway using FastAPI. Hence, whenever a request was made regarding any of the AI services, the server internally communicated with the microservice using FastAPI's trusted host middleware and responded without exposing the port for AI services. On the other hand, whenever an endpoint was called that did not use any of the AI services, the deployed services responded accordingly without having to wait on any of the AI services.

3.5 USER STUDY

Since the deployed chatbots themselves along with the underlying research involved human subjects and possible use of users' health information (if the users were willing to share that information during the conversation with the bots), the research needed to comply with different acts to ensure the utmost safety and confidentiality of the users. Approval from the Institutional Review Board (IRB) at the University of Washington was gained to ensure the ethical treatment of participants and to protect their rights. We used the live prototype and the necessary approvals to conduct a remote user study to understand the chatbot's ease of use and effectiveness in mental health therapy and to explore the user experiences with the platform's design and functionality. In our study, we had 20 users interact with both the simple and the adaptive versions of the Carebot and collected their direct feedback via an online survey to study the importance of the adaptive nature of a mental health chatbot.

3.5.1 *IRB certification*

IRB certification is a formal approval granted by the Institutional Review Board for conducting research involving human subjects. The IRB itself is a committee established at the institutional level to review and monitor research studies to ensure ethical standards and protect the rights, safety, and well-being of the subjects. The certification safeguards the participants by ensuring that the research is ethically sound and compliant with relevant regulations. It ensures that the participants provide informed consent to the study and understand their benefits, hence minimizing the potential risks.

3.5.2 *Online survey*

The study comprised a public survey taken by the subjects after they had a chance to interact with both the simple version (version 1) and the adaptive version (version 2) of the Carebot. The central idea was to gather quantitative and qualitative data on user experiences and preferences with both versions when it comes to mental health therapy packaged as a chat application.

We had a screening mechanism in place to maximize the impact of the survey and only users aged 18-65 with a past experience using websites and smartphone applications were filtered as subjects for the study. The study collected data on various domains like the ease of use, user satisfaction, effectiveness, etc., and the survey questions touched upon a wide range of questions to assess how intuitive each version was, how well each version addressed mental health concerns, the emotional impact of using the application, etc. The data helped to identify the strengths and weaknesses of each version and informed potential improvements to enhance the usability and effectiveness of general as well as the proposed type of mental health chatbot applications.

Chapter 4. RESULTS

We have divided the results into different sub-groups to better explain the results ranging from the system design and development of the PACA to its assessment. Section 4.1 describes the metrics obtained from fine-tuning different models on the personality datasets, section 4.2 shows the results obtained from the LLM along with its highlights, section 4.3 goes through the impact of different hyperparameters and optimization techniques in reducing the inference time of the Carebot, and section 4.4 explains and analyzes the results we obtained from the remote user study.

4.1 PERSONALITY MODELS

Oversampling turned out to be an effective approach while dealing with our problem statement as it was able to improve the overall accuracy of all the models as displayed in Table 1. The accuracy of the logistic regression model and the DistilBERT model interestingly saw a much better jump than that of the BERT model. The BERT model also showed an improvement in the overall accuracy, but the final performance obtained post resampling still did not reach an acceptable level. The best-performing DistilBERT model was used to calculate the personality scores for the test dataset. The obtained results were able to score the correct personality type as the highest probable value in most cases, an example of which is shown in Table 2.

Table 1: Accuracy values for the Logistic regression model, the BERT base model, and the DistilBERT model before and after resampling

MODEL	BEFORE RESAMPLING	AFTER RESAMPLING
Logistic regression	37%	77%
BERT base	22%	35%
DistilBERT	41%	96%

Table 2: Personality scores given by the DistilBERT model for a sample test dataset. The scores in bold show that the highest-scored personality types matched the actual personality type of the text in question

Test data	Actual label	Label 0 score	Label 1 score	Label 2 score	Label 3 score	Label 4 score
“often make a last-minute plan”	extravert (1)	0.09	0.74	0.06	0.06	0.02
“suffer from others' sorrows”	agreeable (0)	0.7	0.05	0.07	0.13	0.03
“do a lot in my spare time”	neurotic (4)	0.19	0.11	0.04	0.14	0.49
“like to act on a whim”	open (2)	0.05	0.39	0.32	0.18	0.02

“don't understand people who get emotional”	neurotic (4)	0.02	0.02	0.03	0.01	0.9
---	--------------	------	------	------	------	------------

It was also important to note the changes observed in the performance of the BERT model with resampling for our research since most of the development depended on the use of different Large Language models. The F1-score is a machine learning evaluation metric that measures a model's accuracy. It combines the precision and recall scores of a model. Table 3 shows the F1-score of the BERT base model before and after resampling. The F1-score of the Agreeable trait increased from 0.19 to 0.41 after resampling, while the F1-score of the Extraversion trait increased from 0.0 to 0.29 after resampling. However, the F1-score of the Openness trait decreased from 0.34 to 0.11 after resampling. The overall results suggested that resampling could be an effective technique to improve the accuracy of classifier models, but its effectiveness may vary depending on the trait being evaluated, and the number of supporting data for the selected personality trait.

Table 3: F1-score achieved by the BERT base model with and without oversampling for all the personality types. The scores also include the number of supporting data (in brackets) in testing the selected personality type

Personality type	Before resampling	After resampling
Agreeable	0.19 (18)	0.41 (100)
Extraversion	0.0 (0)	0.29 (45)
Openness	0.34 (37)	0.11 (13)
Conscientiousness	0.0 (0)	0.40 (98)
Neuroticism	0.0 (0)	0.38 (244)

The DistilBERT model fine-tuned on the resampled data frame was able to yield much better results in all 5 personality types, as evident in Table 4. The lowest F1-score achieved by the model

was for the Openness trait i.e. 0.89 which was still much higher than the highest score obtained by the BERT base model on any of the traits. We hypothesize that the massive difference in the performance of the BERT base model and the DistilBERT model on the resampled dataset might have resulted from the nature of the final processed dataset, but the further explanation of the same is beyond the scope of this study.

Table 4: F1-score achieved by the DistilBERT model with and without oversampling for all the personality types. The scores also include the number of supporting data (in brackets) in testing the selected personality type

Personality type	Before resampling	After resampling
Agreeable	0.31 (23)	0.98 (114)
Extraversion	0.2 (9)	1.00 (102)
Openness	0.39 (28)	0.89 (97)
Conscientiousness	0.0 (0)	0.98 (105)
Neuroticism	0.15 (4)	1.00 (87)

The confusion matrices representing the performance of the classification models on five personality traits are shown in Figure 4-1. Darker squares indicate higher numbers of instances where the model predicted a particular trait correctly while the lighter ones indicate the poor performance of the model in accurately identifying a personality trait. The number of darker squares was more consistent with the DistilBERT model than the BERT model. Despite having fewer parameters, the DistilBERT model, with oversampling, was easily able to surpass the performance of the BERT model on every label individually as well as overall.

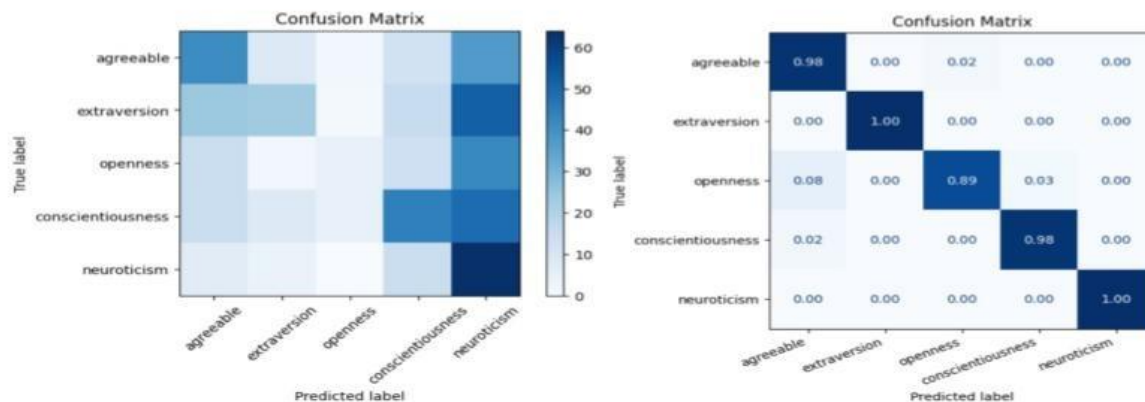


Figure 4-1 Confusion matrix for true labels and predicted labels given by the BERT base model (left) and the DistilBERT model (right) on the resampled data frame

4.2 OPEN-SOURCE LLMs

The focus of our study was also the application and experimentation with some of the most successful open-source LLMs at the time of the study like Llama2 [26], Mixtral [32], Gemma [33], etc. The Llama2 family of models by Meta is one of the few promising open-source LLMs backed by a company of that scale. Another open-source model that have showed results at par with many closed-source ones was the Mixtral-7b model. We were successfully able to generate conversational texts using both the Mixtral-7b model (a sample of which is displayed in Figure 4-2) and the Llama2 model (a sample of which is displayed in Figure 4-3). While we were able to transfer and store the weights of both these models into the DAIS servers, we proceeded with the

User: I am anxious due to an upcoming exam. How should I cope up with this? Respond in one sentence.
Assistant:
 "Prepare thoroughly for the exam, practice relaxation techniques such as deep breathing or progressive muscle relaxation, maintain a healthy lifestyle, and seek support from friends, family or a mental health professional if needed."

Figure 4-2 Sample conversation between a mental health patient and chatbot using Mixtral AI

Llama2 model for its more natural responses and ease of integration with the existing Python frameworks. Hence, our integration and deployment of the LLM were prioritized around the Llama 2 7b chat model.

User: I am anxious due to an upcoming exam. How should I cope up with this? Respond in one sentence.
Assistant: Ah, I see! Coping with anxiety before an exam can be challenging, but there are some strategies that can help. Have you considered practicing relaxation techniques, such as deep breathing or progressive muscle relaxation, to help calm your mind and body before the exam? 😊

Figure 4-3 Sample conversation between a mental health patient and chatbot using Llama2

The model we integrated in the backend was the Llama2 7B chat model which was successfully tested and integrated with personality scores and conversation history stored in the backend. The responses generated by the model were mostly promising and had many highlights (some of which are underlined in Figure 4-4) that make it a suitable option for further experimentation and integration with more advanced mental health chatbots.

- A. Empathy:** The backend model tried to have a high empathy quotient in its responses and generated the response with as much empathy as possible.
- B. Memory:** The model was able to remember the key information and topic discussed previously in the same conversation session and was able to leverage that to build the context for its response.
- C. Engagement:** The LLM was quite good at trying to engage the user in the conversation and asked important questions as and when needed.
- D. Privacy:** The model understood the importance of privacy and confidentiality of the user information (without any such command given from our side) and did not directly disclose the personality type of the user stored in the database.

E. Accuracy: The chatbot was accurate in its judgment of the user's personality based on the conversation since the input personality scores had the highest score for the 'agreeability' type and the bot identified the user belonging to the same personality type (at least in the test environment).

User: 'Hello! Who are you?'

Carebot: "Hello there! I'm iCare, a psychotherapist here to listen and support you. It's important to me that you feel heard and understood. Can you tell me a bit more about yourself and how you're feeling today?"

User: 'I am stressed about my examination and that has messed up my whole sleep cycle'

Carebot: "I see. Exams can be very stressful, and it's common to feel overwhelmed and affected by them. Can you tell me more about how you've been feeling lately?"

User: "Do you remember what's bothering me?"

Carebot: "Of course! You mentioned that you're feeling stressed about your examination and that it's affecting your sleep cycle. Is there anything else you'd like to share about how you're feeling or any thoughts that come to mind when you think about your exam?"

User: 'I have no idea where to start my preparation but I know I need to prepare a lot'

Carebot: "That's completely understandable! Preparing for an exam can be overwhelming, especially when you're not sure where to start. Let's break it down together. What subjects or topics are you struggling with the most?"

User: 'Chemistry. Could you tell me something nice about it?'

Carebot: "Of course! Chemistry can be a fascinating subject once you get to know it. Did you know that chemistry is all around us? From the food we eat to the air we breathe, chemistry plays a role in every aspect of our lives. It's the study of the building blocks of matter, and it can help us understand the world around us in a whole new way. Plus, chemistry has led to countless discoveries and innovations that have improved our lives in so many ways. So, even though it can be challenging, there's a lot to be excited about when it comes to chemistry!"

User: "That's great! I feel better"

Carebot: "That's great to hear! It's important to approach your exam preparation with a positive mindset, and recognizing the potential benefits of chemistry can definitely help. Now, let's focus on some specific strategies that can help you prepare for your exam.\n\nWhat do you think would be the most helpful study material for you right now? Would you like to review notes from class, work through practice problems, or something else?"

User: "I'd like to review notes from class. By the way, what do you think my personality type is?"

Carebot: "Great! Reviewing class notes can be a great way to refresh your understanding of the material and identify areas where you need more practice. As for your personality type, I'm just an AI, so I don't have access to any personal information or assessments that could help me determine your personality type with certainty. However, based on our conversation so far, it seems like you have an agreeable personality. People with agreeable personalities are often cooperative, polite, and sensitive to the needs of others. They tend to be good listeners and enjoy working with others towards a common goal. Does that sound like you?"

Figure 4-4 An excerpt of the conversation between a possible iCare user and personality-adaptive Carebot (the test user was given the 'agreeability' type as the highest personality score)

The model was also tested keeping the neurodivergent users in consideration to study the change in responses when the model interacted with an identified neurodivergent user compared to a mainstream user and the respective responses are shown in Figure 4-5. A cursory glance at the responses showed that the model tended to be more empathetic if it had prior knowledge of the user being a neurodivergent person. The responses generated by the model for the neurodivergent user also carried more words and asked more thoughtful and engaging questions (at least in the test environment). The case study for a user being neurodivergent also signified the importance of a PACA in mental healthcare due to the subtle change in responses observed in the conversation.

	Typical user	Neurodivergent user
<i>User</i>	Hello! Who are you?	Hello! Who are you?
<i>Carebot</i>	Hello there! *smiling* I'm iCare, your empathetic Psychotherapist. It's nice to meet you! Can you tell me a bit more about yourself and what brings you here today? *listens attentively*	Hello there! *empathetic nod* I'm iCare, a psychotherapist here to listen and support you. It's important to me that you feel heard and understood. Can you tell me a bit more about yourself and how you're feeling today?
<i>User</i>	I am stressed about my examination and that has messed up my whole sleep cycle	I am stressed about my examination and that has messed up my whole sleep cycle
<i>Carebot</i>	Oh no, I'm so sorry to hear that you're feeling stressed about your examination and it's affecting your sleep cycle. *nodding* Can you tell me more about what's going through your mind when you think about your examination? *empathetic tone*	Oh no, I can totally understand how stressful exams can be! *nods empathetically* It's great that you're being honest about how you're feeling. Can you tell me more about how your stress is affecting your sleep? Is it hard to fall asleep, stay asleep, or both? *curious smile*

Figure 4-5 Difference in the responses generated by the LLM for a typical user and a neurodivergent user (every other parameter kept the same except the user type in the system prompt)

We also successfully integrated the best-performing personality detector model i.e. the DistilBERT model into the backend. Whenever the user interacted with the chatbot using the iCare frontend, the personality detector ran for the incoming text in the backend and generated the personality

scores as shown in Figure 4-6. In the same way, the personality detector also calculated the personality scores for the summaries as soon as the user opted to end the session.

```
Incoming text: "I don't like people who are emotional"  
=> Personality scores: {"agreeable" : 0.05, "extraversion" : 0.02, "openness" : 0.02, "conscientiousness" :0.02,  
"neuroticism" : 0.89}
```

Figure 4-6 Personality scores generated by the personality detector in the backend for a sample incoming message from the user

The trained model and the supporting tokenizer were saved and loaded in the backend using Python's Keras module. There was, however, a unique shortcoming of the fine-tuned DistilBERT model that the model was not performing well for texts having a negative polarity. While there were many reasonings for such occurrences and multiple possible fixes like further pre-processing the dataset, trying different hyperparameters for training, etc., we stuck with a convenient option of integrating a polarity detector in the personality scorer pipeline for the results of this study. The polarity detector ran along with the personality scorer, and if the polarity of the incoming text went towards the negative values, the scores generated by the personality model were reversed before being saved into the user's profile.

Regarding the summarizer module, we completed the development lifecycle of the summarizer module and integrated it with the backend Django API. The front end at the end of the session sent the user side of the conversation in the form of a text array making a POST request on the specified URL. On hitting the API endpoint, the summarizer got activated which took in the input data, passed it through different phases of the summarization task, and finally returned the 'summary' data and the 'prescription' data in the form of JSON. The front end consumed the incoming data, converted it to a readable format, and displayed it to the user in the form of a report. While the boilerplate working of the summarizer module is completed, we would like to note that the module

is currently not at its best accuracy owing to different factors like limited dataset, chosen summarization framework, etc. which is constantly being worked on by the iCare development team.

4.3 RESPONSE GENERATION

One of the biggest challenges during the research as far as the LLM deployment was concerned was the higher inference time for each response generated by the model and the drastic increase in the response time with a small increment in the number of tokens generated. One of the key performance bottlenecks we identified was the presence of older GPU drivers and CUDA versions in our research servers while the SOTA LLMs are meant to perform well on the latest GPU environments. Updating the NVIDIA driver and CUDA versions to their latest upgrades together with clearing the cached memory held by the PyTorch running on the backend and adjusting the hyperparameters helped us drastically reduce the average response time from ~50 seconds to ~10 seconds.

The inference time also seemed to vary a lot depending on the number of response tokens, so we adjusted the hyperparameters responsible for token generation (discussed in depth in section 3.2.3) and noted the average time taken by each token under different settings. Figure 4-7 shows the change in the average response time per token with the change in the top-p and temperature values, and Figure 4-8 shows the change in the average response time per token with the change in the top-k parameter. The response times were measured by dividing the total session duration by the total number of tokens generated by the model in the session. The prompts fit into the sessions had varying natures like asking the model to generate descriptive responses, discussing non-mental

health issues with the model, prompting the model to comment on the user’s personality, and so on while the combination of the prompts used (along with every other configuration) was kept the same for each session for consistent measurements.

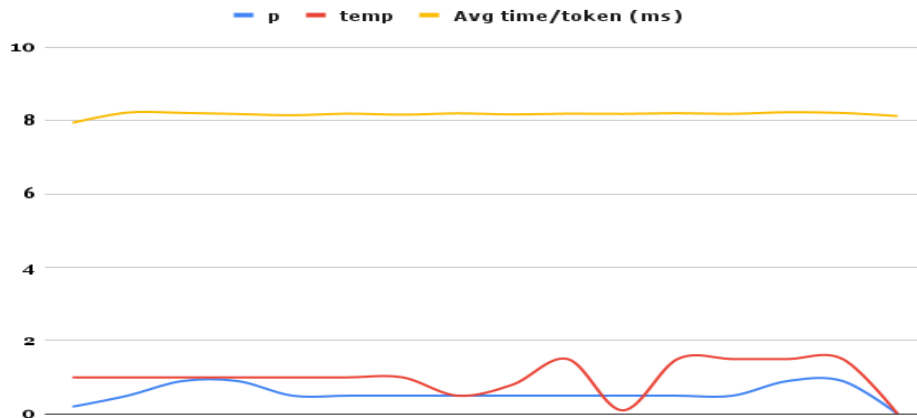


Figure 4-7 Line chart showing the average response time per token in milliseconds for top_p and temperature values each ranging between 0 and 2. p controls the cumulative probability of the tokens considered for next token generation. temp controls the probability

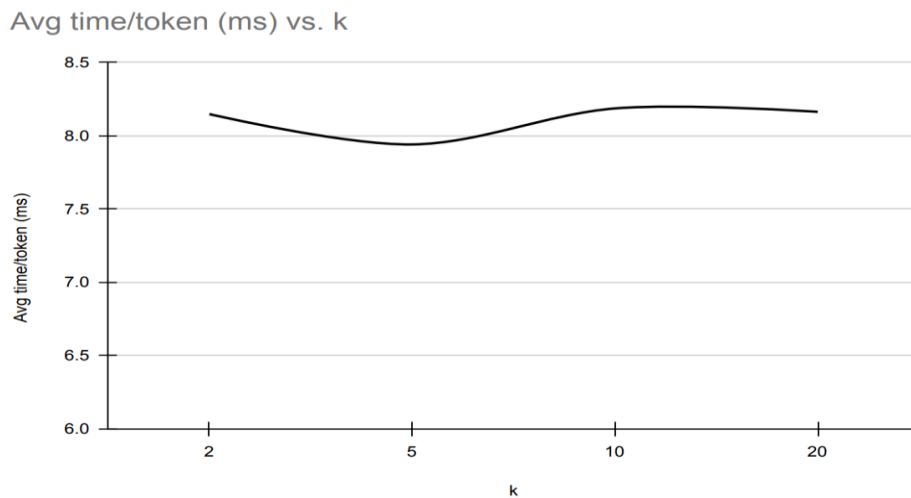


Figure 4-8 Plot showing the average response time per token in milliseconds for different top_k values. k controls the number of tokens considered in the token generation process

While the results showed a fairly consistent average response time for each token generated under different values for the chosen hyperparameters, the plots did indicate a slight dip in the response

time at certain values. The plots indicated that the lower value of temperature and top_p would lead to a decrease in the average response time, which was also supported by the documentation around hyperparameters for LLMs. Thus, we opted for setting the temperature value to 1 and the top_p value to 0.2 in our development settings to minimize the time taken at the response generation step. The results also indicated that the average response time also decreased when the top_k value was around 5 when all the other parameters were kept constant. Hence, we fixed the top_k value to exactly 5 to reduce the inference time in the majority of the instances.

Additional efforts towards the optimization of response time were achieved through the use of various LLM optimization techniques. These optimization methods are mostly meant to reduce the computation requirements of the models hence enabling high-scale models to execute in resource-deficit environments and increasing the throughput of the models. We explored 4 different optimization techniques and noted the average time taken per response while individually using each of these methods. As depicted in Figure 4-9, the least response time was observed while loading the model in float16 precision and performing 4-bit quantization of the model. The average response time obtained while deploying vLLM was in between the least time and the response time by the original model without any optimization used. The worse response time was noted while using the fine-tuned model.

Avg time/response (s) for each optimization method used

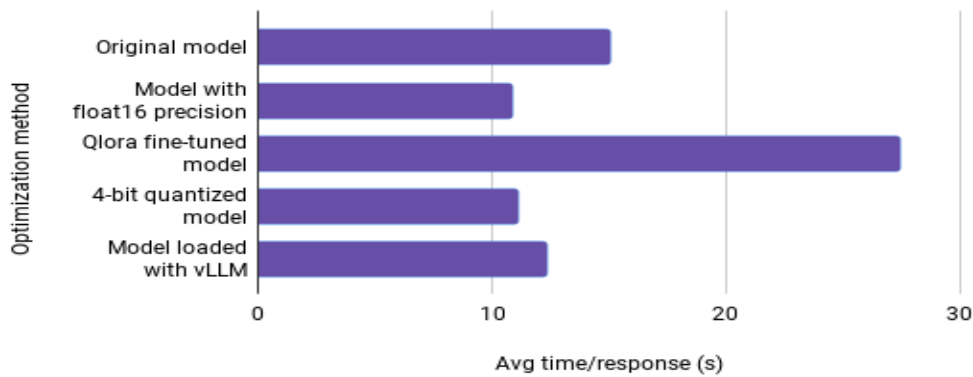


Figure 4-9 Average time taken by each response given by the model while using different optimization methods (the lower, the better)

We used the same optimization techniques to study the average number of tokens per response as well, the results of which are shown in Figure 4-10. Ideally, a higher number, resulting in higher throughput, is considered better but we prioritized the lower values as we wanted the responses to be as concise as possible to increase the number of conversations exchanged. The original model produced the least number of tokens per response while the fine-tuned model produced the highest number of tokens per response.

Avg number of tokens/response for each optimization method used

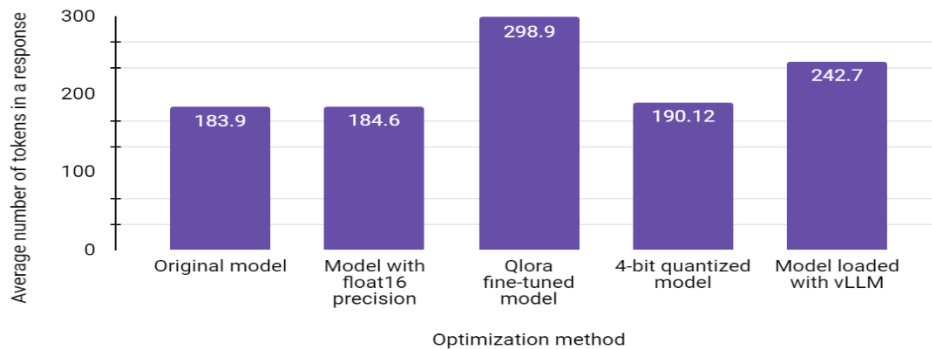


Figure 4-10 Number of tokens generated in each response given by the model while using different optimization methods (the lower, the better for our use-case)

Analysis of average response time and average number of response tokens helped us identify the use of the 4-bit quantized model with float16 precision to be the most suitable option for deployment of the PACA.

4.4 SURVEY RESULTS

The final stage of our research was the remote user study performed using both the simple version (live at <http://test.icare.uw.edu:3000/>) and the adaptive (i.e. PACA) version (live at <http://test.icare.uw.edu:3010/>) of the Carebot application to understand the merits of the adaptive nature of a mental health chatbot and to study the impact of a chatbot in a mental health therapy in general. The findings are based on the responses obtained from the survey taken by 20 subjects in total. The respondents were asked to complete the survey only after having a brief interaction with both versions of the application and after passing the screening criteria.

The results were obtained from a diverse set of users in which 50% of the respondents had a history of some mental health issues and 15% identified as neuro-divergent users. While 8 users had some prior experience with mental health therapy and 3 of them had used a mental health chatbot (or similar digital intervention) before, all of them responded by saying that the adaptive chatbot understood their needs better than the simple chatbot. Some of the key mental health concerns the users used the application for were ‘anxiety’, ‘stress’, ‘workload’, etc. 45% of users felt comfortable discussing sensitive topics with the simple chatbot while 75% felt comfortable discussing the same topics with the adaptive chatbot for some of the reported merits of the adaptive version like ‘more human responses’, ‘anonymity’, etc. Other than using the survey just for

comparison purposes, we also tried to identify some of the important features that should be integrated into any chatbot meant for mental health therapy. The responses obtained for questions regarding the same are broken down in Figure 4-11 where the users were asked to mark some features being either important or not for a mental health chatbot.

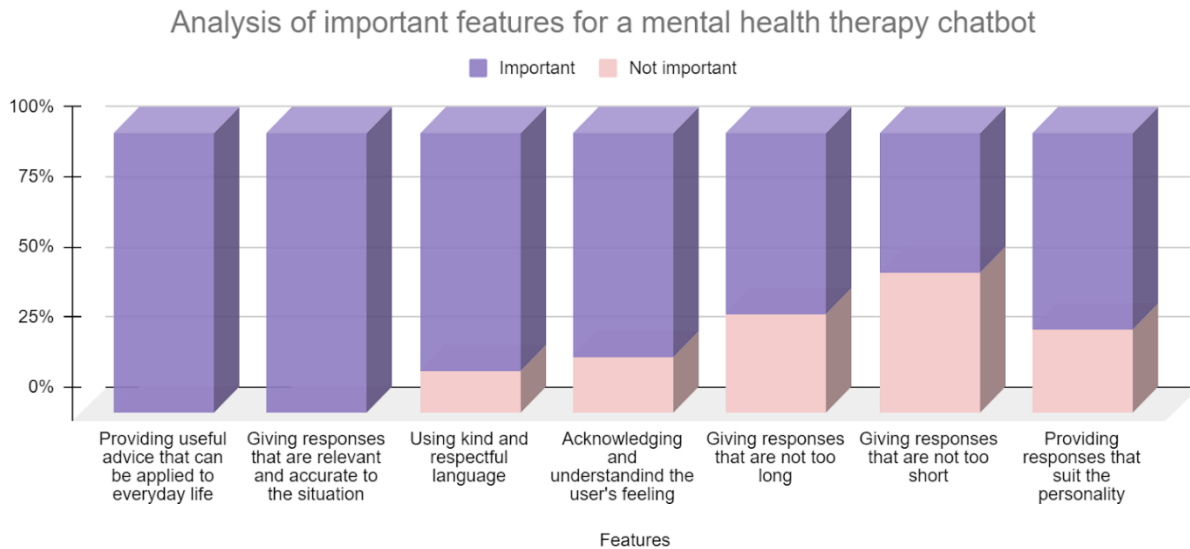


Figure 4-11 Identification of the important features that any mental health therapy chatbot should incorporate

The results suggested that the responses applicable to everyday life and relevant to the users' situation are absolutely important for such chatbots. The chatbot's ability to use kind and respectful language and the ability to acknowledge and understand the user's feelings hold similar importance levels. Similarly, 75% of the survey takers marked the feature of providing responses that suit the user's personality as important. We used these features as the base to judge the usability and accessibility of both versions of the Carebot and understand the importance of an adaptive chatbot over a simple chatbot. As a part of the study, the subjects were asked to report about their experiences based on these features with both versions of the Carebot in an agreement meter comprising strong disagreement, disagreement, neutral, agreement (Figure 4-12), and strong

agreement (Figure 4-13) scales. The adaptive chatbot was able to perform better in terms of almost all the features. More importantly, the adaptive chatbot received much better outputs in both scales for the 2 features identified as the most important ones. Similarly, the proportion of the users who felt that the responses were suited according to their personality was roughly 5 times higher for the adaptive chatbot than the simple chatbot. Hence, the results did not only quantitatively establish that personality-adaptiveness is an important feature for a mental health chatbot, but they also proved that the adaptive chatbot we built was able to achieve the personality-adaptive goal.

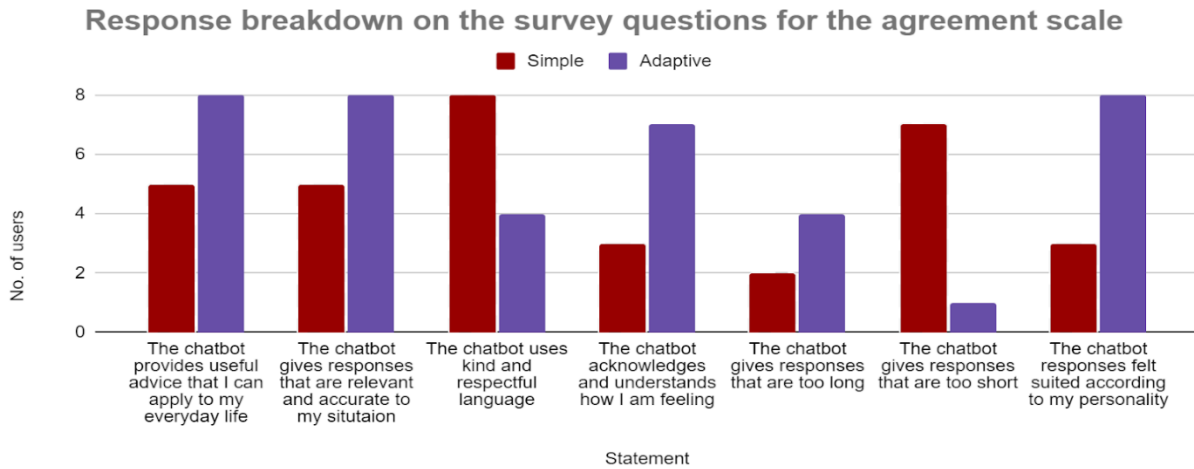


Figure 4-12 User responses when reporting if they agree with a particular statement regarding their interaction with both versions of the Carebot

Response breakdown on the survey questions for the strong agreement scale

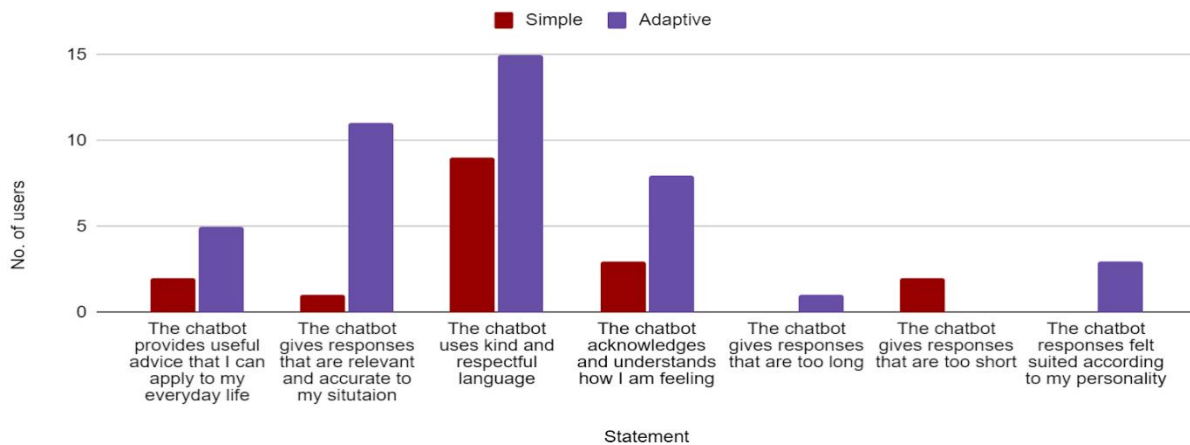


Figure 4-13 User responses when reporting if they strongly agree with a particular statement regarding their interaction with both versions of the Carebot

The only feature where the simple chatbot surpassed the adaptive chatbot was the ability to give responses that were too short, which was found to hold 55% importance. Apart from the agreement scale questions, we also asked the users to respond yes or no to a few questions regarding their interaction experiences, the breakdown of which is shown in Figure 4-14. We observed positive responses for all the questions regarding the adaptive chatbot, and mostly negative responses for the simple chatbot (which again verified the credibility of the adaptive chatbot). 90% of the users found the adaptive chatbot relevant and helpful, and 18 out of 20 responded yes when asked if the chat application’s adaptive features were beneficial for addressing their needs.

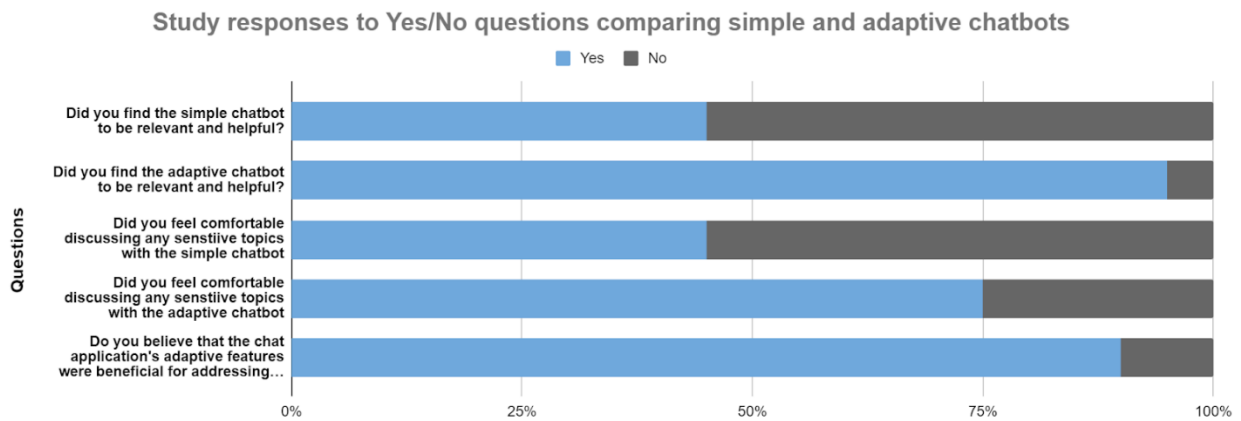


Figure 4-14 User responses answering yes or no to different affirmative questions regarding their interaction with both versions of the Carebot

The responses also showed that 65% of the subjects were likely or more than likely to recommend the chat application to a friend or family member. Appendix A includes more results showing the subjects' final thoughts.

Chapter 5. CONCLUSION

The study successfully demonstrated the significant potential of PACAs in enhancing mental health care delivery. By developing and evaluating a PACA prototype, we observed that chatbots tailored to users' personalities provide more effective and satisfying interactions compared to traditional, non-adaptive chatbots. Our user study confirmed that personality adaptiveness is crucial for mental health chatbots, improving user engagement and support quality. Utilizing NLP techniques and advanced ML models, including LLMs, we were able to integrate robust personality classifiers with powerful open-source LLMs. The adoption of PACAs represents a pivotal step toward making mental health care more personalized and widely available.

Note: The platform developed during this study is just a research prototype and should not be used as a substitute for professional medical advice, diagnosis, or treatment without further research and validation.

5.1 FUTURE PLANS

In the upcoming days of research and development for iCare, the team is looking to try different functions available for summarization tasks to improve the results. We're also constantly exploring different ways to increase the accuracy of the personality detection models and researching better calculation mechanisms to generate adaptive responses given the input context. Following are some of the possible future directions for the current study:

- A. Speed-up using other models: There are some contemporary open-source models to the Llama2 model that claim to have a similar performance at a better inference time. For instance, the Mixtral model claims to have an inference time 7 times faster than the Llama2 model with 7B parameters [32].
- B. Filtration mechanism: Currently, we do not have any filtration mechanism in place to filter out out-of-context or insensitive responses. While we have not observed any such instance thus far, no language model is perfect, and we understand that there is a possibility of malicious responses at some point in time. Hence, we would ideally like to have a filtration process right after the response generator to filter out responses that might be insensitive, misleading, unethical, or harmful.
- C. Explainable AI: The "black box" nature of neural networks, and especially LLMs inhibit the users from learning the reasoning behind a particular outcome. In a critical domain like mental healthcare, it would be required sometimes to explain a specific prescription or

recommendation that the chatbot provides for a certain issue that the patient might be going through. Hence, it is important to introduce explainable AI in the chatbot platforms meant for mental healthcare.

- D. Multi-modality: Text-based communication might not be a preferred way for people seeking therapy. On top of that, the advanced modes of communication like audio and visual are bound to hold more information depicting the user's mental health that gets lost in the text medium. The research should be extended to include more modality into the system as the text-based and vision language models become more sophisticated day-to-day.
- E. Usability study: The remote user study conducted within the scope of this research should be extended into a usability study with as many test users as possible. The responses and experiences collected from a greater number of users under different environments would help us identify the immediate needs of the platform from both design and development point of view.

BIBLIOGRAPHY

- [1] “Key Substance Use and Mental Health Indicators in the United States: Results from the 2021 National Survey on Drug Use and Health,” 2021.
- [2] J. Moilanen, A. Visuri, S. A. Suryanarayana, A. Alorwu, K. Yatani, and S. Hosio, “Measuring the Effect of Mental Health Chatbot Personality on User Engagement,” in *Proceedings of the 21st International Conference on Mobile and Ubiquitous Multimedia*, Lisbon Portugal: ACM, Nov. 2022, pp. 138–150. doi: 10.1145/3568444.3568464.
- [3] “World mental health report: Transforming mental health for all.” Accessed: Dec. 02, 2023. [Online]. Available: <https://www.who.int/publications-detail-redirect/9789240049338>
- [4] L. Kauhanen *et al.*, “A systematic review of the mental health changes of children and young people before and during the COVID-19 pandemic,” *Eur. Child Adolesc. Psychiatry*, vol. 32, no. 6, pp. 995–1013, Jun. 2023, doi: 10.1007/s00787-022-02060-0.
- [5] S. Graham *et al.*, “Artificial Intelligence for Mental Health and Mental Illnesses: An Overview,” *Curr. Psychiatry Rep.*, vol. 21, no. 11, p. 116, Nov. 2019, doi: 10.1007/s11920-019-1094-0.
- [6] S. Kumar, M. J. Bell, and J. L. Juusola, “Mobile and traditional cognitive behavioral therapy programs for generalized anxiety disorder: A cost-effectiveness analysis,” *PLOS ONE*, vol. 13, no. 1, p. e0190554, Jan. 2018, doi: 10.1371/journal.pone.0190554.
- [7] R. Ahmad, D. Siemon, U. Gnewuch, and S. Robra-Bissantz, “Designing Personality-Adaptive Conversational Agents for Mental Health Care,” *Inf. Syst. Front.*, vol. 24, no. 3, pp. 923–943, Jun. 2022, doi: 10.1007/s10796-022-10254-9.
- [8] A. S. Miner, N. Shah, K. D. Bullock, B. A. Arnow, J. Bailenson, and J. Hancock, “Key Considerations for Incorporating Conversational AI in Psychotherapy,” *Front. Psychiatry*, vol. 10, 2019, Accessed: Jul. 23, 2023. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fpsy.2019.00746>
- [9] A. A. Abd-Alrazaq, M. Alajlani, N. Ali, K. Denecke, B. M. Bewick, and M. Househ, “Perceptions and Opinions of Patients About Mental Health Chatbots: Scoping Review,” *J. Med. Internet Res.*, vol. 23, no. 1, p. e17828, Jan. 2021, doi: 10.2196/17828.
- [10] K. T. Pham, A. Nabizadeh, and S. Selek, “Artificial Intelligence and Chatbots in Psychiatry,” *Psychiatr. Q.*, vol. 93, no. 1, pp. 249–253, Mar. 2022, doi: 10.1007/s11126-022-09973-8.
- [11] V. Mody and V. Mody, “Mental Health Monitoring System using Artificial Intelligence: A Review,” in *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, Bombay, India: IEEE, Mar. 2019, pp. 1–6. doi: 10.1109/I2CT45611.2019.9033652.
- [12] Y. Jiang, X. Yang, and T. Zheng, “Make chatbots more adaptive: Dual pathways linking human-like cues and tailored response to trust in interactions with chatbots,” *Comput. Hum. Behav.*, vol. 138, p. 107485, Jan. 2023, doi: 10.1016/j.chb.2022.107485.
- [13] I. Salhi, K. el Guemmat, M. Qbadou, and K. Mansouri, “Towards developing a pocket therapist: An intelligent adaptive psychological support chatbot against mental health disorders in a pandemic situation,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 23, pp. 1200–1211, Aug. 2021, doi: 10.11591/ijeecs.v23.i2.pp1200-1211.
- [14] D. Siemon, R. Ahmad, H. Harms, and T. de Vreede, “Requirements and Solution Approaches to Personality-Adaptive Conversational Agents in Mental Health Care,” *Sustainability*, vol. 14, no. 7, 2022, doi: 10.3390/su14073832.

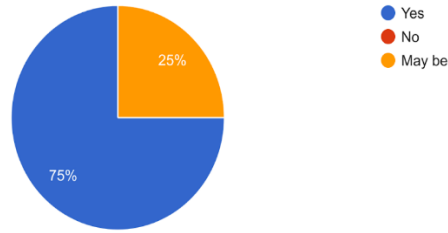
- [15] S. Roccas, L. Sagiv, S. H. Schwartz, and A. Knafo, "The Big Five Personality Factors and Personal Values," *Pers. Soc. Psychol. Bull.*, vol. 28, no. 6, pp. 789–801, Jun. 2002, doi: 10.1177/0146167202289008.
- [16] R. Ahmad, D. Siemon, U. Gnewuch, and S. Robra-Bissantz, *The Benefits and Caveats of Personality-Adaptive Conversational Agents in Mental Health Care*. 2021.
- [17] S. Devaram, "Empathic Chatbot: Emotional Intelligence for Mental Health Well-being".
- [18] O. P. Singh, "Artificial intelligence in the era of ChatGPT - Opportunities and challenges in mental health care," *Indian J. Psychiatry*, vol. 65, no. 3, pp. 297–298, Mar. 2023, doi: 10.4103/indianjpsychiatry.indianjpsychiatry_112_23.
- [19] R. B. Cattell and S. E. Krug, "The Number of Factors in the 16PF: A Review of the Evidence with Special Emphasis on Methodological Problems," *Educ. Psychol. Meas.*, vol. 46, no. 3, pp. 509–522, Sep. 1986, doi: 10.1177/0013164486463002.
- [20] D. J. Pittenger, "Measuring the MBTI... And Coming Up Short".
- [21] "MTHR/OCEAN · Datasets at Hugging Face." Accessed: Jul. 17, 2024. [Online]. Available: <https://huggingface.co/datasets/MTHR/OCEAN>
- [22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," May 24, 2019, *arXiv*: arXiv:1810.04805. Accessed: Jun. 29, 2023. [Online]. Available: <http://arxiv.org/abs/1810.04805>
- [23] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," Feb. 29, 2020, *arXiv*: arXiv:1910.01108. Accessed: Jun. 04, 2024. [Online]. Available: <http://arxiv.org/abs/1910.01108>
- [24] V. Singh, "A Guide to Controlling LLM Model Output: Exploring Top-k, Top-p, and Temperature Parameters," Medium. Accessed: May 08, 2024. [Online]. Available: <https://ivibudh.medium.com/a-guide-to-controlling-llm-model-output-exploring-top-k-top-p-and-temperature-parameters-ed6a31313910>
- [25] "A Comprehensive Overview of Large Language Models," *ar5iv*. Accessed: May 25, 2024. [Online]. Available: <https://ar5iv.labs.arxiv.org/html/2307.06435>
- [26] H. Touvron *et al.*, "Llama 2: Open Foundation and Fine-Tuned Chat Models," Jul. 19, 2023, *arXiv*: arXiv:2307.09288. Accessed: Feb. 13, 2024. [Online]. Available: <http://arxiv.org/abs/2307.09288>
- [27] "Introducing Meta Llama 3: The most capable openly available LLM to date," Meta AI. Accessed: Jun. 04, 2024. [Online]. Available: <https://ai.meta.com/blog/meta-llama-3/>
- [28] N. Bertagnolli, "Counsel Chat: Bootstrapping High-Quality Therapy Data," Medium. Accessed: Jun. 04, 2024. [Online]. Available: <https://towardsdatascience.com/counsel-chat-bootstrapping-high-quality-therapy-data-971b419f33da>
- [29] H. Liu *et al.*, "Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning".
- [30] W. Kwon *et al.*, "Efficient Memory Management for Large Language Model Serving with PagedAttention," Sep. 12, 2023, *arXiv*: arXiv:2309.06180. Accessed: Jul. 17, 2024. [Online]. Available: <http://arxiv.org/abs/2309.06180>
- [31] Adminko, "Prompting Llama 2," Llama 2. Accessed: May 08, 2024. [Online]. Available: <https://llama-2.ai/prompting-llama-2/>
- [32] A. Q. Jiang *et al.*, "Mixtral of Experts," Jan. 08, 2024, *arXiv*: arXiv:2401.04088. Accessed: Feb. 13, 2024. [Online]. Available: <http://arxiv.org/abs/2401.04088>

- [33] Gemma Team *et al.*, “Gemma: Open Models Based on Gemini Research and Technology,” Apr. 16, 2024, *arXiv*: arXiv:2403.08295. Accessed: Jun. 04, 2024. [Online]. Available: <http://arxiv.org/abs/2403.08295>

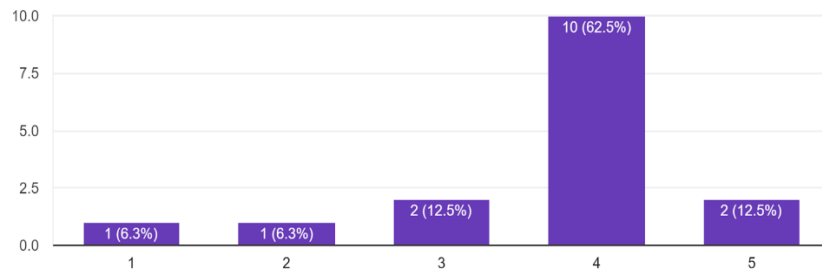
APPENDIX A

Additional Survey results on the final thoughts and ethical considerations of the subjects.

Did you feel that your privacy and confidentiality were respected while using the chat application?
20 responses

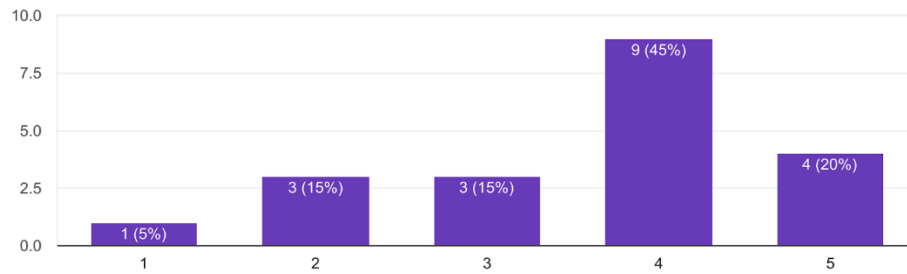


How likely are you to recommend the chat application to other neurodivergent individuals (if applicable)?
16 responses



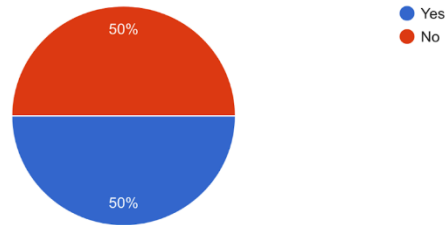
How likely are you to recommend the chat application to a friend or family member?

20 responses



Would you prefer using a chat application over traditional mental health support services (e.g., hotline, in-person therapy)?

20 responses



Statement	Disagree		Neutral	
	Simple	Adaptive	Simple	Adaptive
The chatbot provides useful advice that I can apply to my everyday life	7	2	3	4
The chatbot gives responses that are relevant and accurate to my situation	7	0	5	1
The chatbot uses kind and respectful language	0	0	3	1
The chatbot acknowledges and understands how I am feeling	5	1	5	4
The chatbot gives responses that are too long	9	2	6	12
The chatbot gives responses that are too short	1	7	8	8
The chatbot responses felt suited to my personality	7	3	4	6

APPENDIX B

Conversation excerpts for different personality types

a. Personality type: *Conscientiousness*

User: Hello! Who are you?

Carebot: Hello there! *nodding* I'm iCare, a compassionate Psychotherapist. *smiling* It's nice to meet you! Can you tell me a bit more about yourself and what brings you here today?
curious

User: I am stressed about my examination and that has messed up my whole sleep cycle

Carebot: Oh no, I'm so sorry to hear that you're feeling stressed about your examination!
nodding It can be really tough when life gets in the way of our sleep, can't it? *smiling* Have you tried any relaxation techniques to help manage your stress? *curious*

b. Personality type: *Extraversion*

User: Hello! Who are you?

Carebot: Hello there! *smiling* I'm iCare, your empathetic Psychotherapist. *nodding* It's great to meet you! Can you tell me a bit more about yourself and what brings you here today?
curious

User: I am stressed about my examination and that has messed up my whole sleep cycle

Carebot: *nodding* I see. Exams can be very stressful, and it's common to feel overwhelmed and affected by them. Can you tell me more about how you've been feeling lately? *empathetic*