A

Minor Project Report

On

MENTAL HEALTH ASSISTING AGENT

Submitted to

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE AND TECHNOLOGIES RK VALLEY

On completion of Minor project

Submitted by

Arigala Adarsh R190063

Under the Guidance of

Dr. Penugonda Ravi Kumar

PhD(University of AIZU, Japan), M.E (IISC Bangalore)
Assistant Professor
Department of Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
(catering the Educational Needs of Gifted Rural Youth of AP)
R.K Valley, Vempalli(M), Kadapa (Dist.) 516330

2023 – 2024

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

(A.P.Government Act18of2008)

RGUKT-RK Valley

Vempalli, Kadapa, Andhra Pradesh-516330.

DECLARATION

I am A.Adarsh(R190063) here by declaring that the project report entitled "**MENTAL HEALTH ASSISTING AGENT**" was done under guidance of Dr. P Ravi Kumar is submitted for completion of Minor Project in Computer Science and Engineering during the academic session January 2024 – July 2024 at RGUKT-RK Valley. We also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites are mentioned in the references. To the best of our knowledge, the results embodied in this dissertation work have not been submitted to any university or institute for completion of the Minor Project.

Date:
Place : RK Valley

ARIGALA ADARSH(R190063)

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

(A.P.Government Act18of2008)

RGUKT-RK Valley

Vempalli, Kadapa, Andhra Pradesh-516330.

CERTIFICATE OF PROJECT COMPLETION

This is to certify that the project work titled "**MENTAL HEALTH ASSISTING AGENT**" is a bonafide project work submitted by **ARIGALA ADARSH – R190063** in the department of COMPUTER SCIENCE AND ENGINEERING in completion of **MINOR PROJECT** during academic session **January 2024 – July 2024 at RGUKT-RK VALLEY** carried out under the supervision.

Project Guide
Dr. P Ravi Kumar) PhD(AIZU, Japan),
M.E (IISC Bangalore),
Asst. Prof. in Dept of CSE,
RGUKT-RK Valley.

Head of the Department
Dr. CH. Ratna Kumari
Asst. Prof. in Dept of CSE

ACKNOWLEDGEMENT

I would like to express my deep sense of gratitude & respect to all those people behind the screen who guided, inspired and helped us crown all our efforts with success. We wish to express our gratitude to Dr. P Ravi Kumar for his valuable guidance at all stages of study, advice, constructive suggestions, supportive attitude and continuous encouragement, without which it would not be possible to complete this project.

We would also like to extend our deepest gratitude & reverence to the Director of RGUKT, RK Valley **Dr. A V S S Kumara Swami Gupta** and HOD of Computer Science and Engineering **Dr. CH Ratna Kumari** for their constant support and encouragement.

Last but not least we express our gratitude to our parents for their constant source of encouragement and inspiration for us to keep our morals high.

WITH SINCERE REGARDS

ARIGALA ADARSH - R190063

TABLE OF CONTENTS

ABSTRACT

Project Title:

Personalized Motivational Chatbot

Objective:

The primary objective of this project is to develop a Mental Health Assisting Agent Provides emotional support and motivation to students and individuals dealing with depression. This chatbot will leverage advanced NLP techniques and LSTM-based RNN models to understand user input and deliver personalized responses, aiming to improve mental well-being and provide timely support.

Problem Statement:

Mental health challenges, such as depression, are common among students and individuals, often worsened by academic pressures and life struggles. Conventional support systems may not always be available or adequate, creating a need for innovative solutions that provide ongoing, personalized emotional support and motivation.

Methodology:

Our goal was to understand user needs and gather insights to guide the development of a chatbot that personalizes motivational support experiences.

1. Data Collection

- **Datasets:** Data was collected from reputable online datasets that specialize in mental health and depression. These datasets include user conversations, therapeutic dialogues, and motivational messages.
- **Books:** Techniques and strategies for dealing with depression were sourced from established books in the field of psychology and mental health.

2. Data Analysis

- **Text Data:** Text data was preprocessed using NLP techniques to clean, tokenize, and normalize the text. Sentiment analysis was conducted to categorize the motivational content.

The insights gleaned from this methodology will inform the design and development of the chatbot. The features and functionalities will be prioritized based on user needs and preferences.

Project Scope:

The project scope includes the deliverables, i.e., functionalities and user interface.

Deliverables

- **Functionalities:**
    - Personalized motivational messages based on user input.
    - Real-time emotional support with adaptive responses.
    - Daily positive affirmations and mental health tips.
    - Interactive and engaging user interface.
- **User Interface (UI):** A user-friendly and intuitive interface will be designed to ensure a smooth and engaging support experience using Streamlit.

Significance of Project:

The Mental Health Assisting Agent project addresses the critical need for accessible and personalized emotional support for students and individuals facing depression. By leveraging ANNs and LSTMs in RNN and advanced NLP techniques, the chatbot offers a scalable and effective solution to improve mental well-being and provide continuous motivation.

Tech-Stack and Tools Used:

- **Tech-Stack:**
    - Python
    - TensorFlow/Keras
    - Artificial Neural Networks (ANN)
    - Long Short-Term Memory (LSTM) in Recurrent Neural Networks (RNN)
- ■ **Tools:**
    - Streamlit (for user interface)
    - Visual Studio Code

Conclusion:

This project demonstrates a successful proof of concept for an innovative approach to supporting mental well-being. By delivering personalized motivational messages and real-time emotional support, the chatbot addresses individual needs and promotes better mental health outcomes. This tool has the potential to democratize mental health support, making it more flexible and accessible to a broader audience. This project paves the way for a future where technology offers personalized emotional support empowering individuals to achieve their mental health and motivational goals.

CHAPTER-1

INTRODUCTION

Mental health issues, including depression, are common among students and individuals, often intensified by academic pressures and life challenges. Traditional support systems may not always be accessible or adequate, creating the need for an innovative solution that offers continuous, personalized emotional support and motivation.

1.1 Motivation:

The Mental Health Assisting Agent project addresses the pressing need for emotional support and motivation among students and individuals experiencing depression. Many of these individuals lack accessible resources and guidance to effectively manage their mental health. By harnessing technology to create a chatbot, we aim to provide users with the support they need to navigate their challenges and enhance their well-being. This initiative has the potential to democratize mental health support, fostering emotional resilience and improving overall well-being for a broader audience.

1.2 Objectives:

The primary objective of this project is to design and implement a chatbot that offers personalized, real-time support to users experiencing mental health challenges. By leveraging advanced NLP techniques and LSTM-based RNN models, the chatbot will interpret user inputs to deliver tailored motivational messages and immediate emotional support. Additional objectives include providing daily positive affirmations, creating a user-friendly interface, and offering features that enhance the user experience.

1.3 Features:

- **Tailored Motivational Messages**: Utiizing advanced NLP techniques and LSTM-based RNN models, the chatbot will interpret user input and deliver customized motivational messages. These messages are tailored to encourage and support users based on their current emotional state.
- **Immediate Emotional Support**: The chatbot will offer real-time emotional support to enhance the user experience and address concerns promptly. This feature ensures users receive immediate responses and comfort during challenging times.
- **Daily Uplifting Affirmations**: Users will receive daily positive affirmations and mental health tips from the chatbot. These affirmations are designed to help maintain a positive outlook and foster healthy coping strategies.

- **User-Friendly Interface**: An intuitive and user-friendly interface will be developed using Streamlit to provide a smooth and engaging support experience.
- **Conversation Refresh Option**: Users can refresh the chat to clear previous conversations and start anew, ensuring a fresh and uncluttered.
- .**Previous Prompts for Similar Questions:** The chatbot will offer previous prompts for similar questions that most users have asked. Users can click on these prompts to receive immediate replies without having to type out their questions, streamlining the support process.

1.4 Impact:

The potential impact of this project is substantial. By providing accessible and personalized mental health support, the chatbot can help users manage their emotional well-being more effectively. The integration of advanced NLP techniques and LSTM-based models ensures that the support is both relevant and timely, addressing the unique needs of each user. Ultimately, this project aims to democratize mental health support, making it available to a broader audience and fostering greater emotional resilience and overall well-being.

1.5 Future Scope:

The future scope of the Mental Health Assisting Agent includes several avenues for expansion and improvement:

- **Mood Tracking:** Integrating tools for tracking user moods over time to provide personalized feedback based on emotional trends.
- **Cognitive Behavioral Therapy (CBT) Exercises:** Incorporating practical CBT exercises and techniques to help users manage their thoughts and behaviors effectively.
- **Real-Time Crisis Intervention:** Implementing features for immediate crisis management, including automated alerts to mental health professionals during critical situations.
- **Multilingual Support:** Expanding the chatbot's capabilities to support multiple languages, making it accessible to a wider audience.
- **Enhanced Automation and Analysis:** Using machine learning techniques to improve the chatbot's ability to analyze user interactions and provide more effective support.

CHAPTER – 2

LITERATURE REVIEW

1.Introduction to Chatbots in Mental Health

Chatbots have gained prominence as tools for providing mental health support due to their accessibility, scalability, and ability to offer immediate assistance. They serve as virtual assistants that can engage users in conversations, provide emotional support, and even guide users through therapeutic exercises. The integration of Natural Language Processing (NLP) and machine learning models, such as Long Short-Term Memory (LSTM) networks, has significantly enhanced the ability of chatbots to understand and respond to user inputs effectively.

2. NLP and LSTM in Chatbots

Natural Language Processing (NLP) is a critical component in the development of chatbots, enabling them to understand and process human language. Long Short-Term Memory (LSTM) networks, a type of Recurrent Neural Network (RNN), are particularly effective in handling sequential data and capturing long-term dependencies, making them suitable for tasks involving language understanding and generation. LSTM models have been widely used in various applications, including language translation, text generation, and more recently, in conversational agents.

3. Applications of LSTM in Mental Health Chatbots

LSTM-based models have been employed in mental health chatbots to enhance their conversational abilities and provide personalized support. These models can analyze user input, detect emotional states, and generate appropriate responses. The key advantages of using LSTM in this context include the ability to maintain context over long conversations and the capacity to learn from large datasets of therapeutic dialogues and user interactions.

4. Case Studies and Examples

- **Woebot**: Woebot is a well-known example of a mental health chatbot that uses NLP techniques to provide cognitive behavioral therapy (CBT) to users. It engages users in conversations, offers mood tracking, and delivers personalized therapeutic content based on user interactions.
- **Wysa**: Wysa is another AI-driven chatbot that uses evidence-based therapeutic techniques, including CBT, Dialectical Behavior Therapy (DBT), and mindfulness, to assist users with mental health challenges. It employs machine learning models to understand user inputs and deliver tailored support.
- **Replika**: Replika is an AI chatbot that uses deep learning techniques, including LSTM, to engage users in empathetic conversations. It helps users manage their mental health by providing a platform for expressing thoughts and feelings in a non-judgmental environment.

5. Advantages and Challenges

The use of LSTM in mental health chatbots offers several advantages, including:

- **Personalization**: LSTM models can learn from user interactions and provide responses tailored to individual needs.
- **Context Retention**: LSTM networks can maintain context over long conversations, making interactions more coherent and meaningful.
- **Scalability**: Chatbots can provide support to a large number of users simultaneously, addressing the accessibility issue in mental health care.

However, there are also challenges to consider:

- **Data Privacy**: Ensuring the confidentiality and security of user data is paramount in mental health applications.
- **Ethical Considerations**: The deployment of AI in mental health requires careful consideration of ethical issues, including the potential for misuse and the need for human oversight.
- **Accuracy and Reliability**: Ensuring the chatbot provides accurate and reliable support is crucial, as incorrect responses can have serious implications for users.

6.Future Directions

The development of mental health chatbots using LSTM and other advanced machine learning models is an ongoing area of research. Future directions include:

- **Improved Emotional Understanding**: Enhancing the ability of chatbots to detect and respond to a wider range of emotional states.
- **Integration with Wearable Devices**: Combining chatbots with wearable technology to provide real-time monitoring and intervention.
- **Multilingual Support**: Expanding the capabilities of chatbots to support multiple languages, making them accessible to a broader audience.

7. Conclusion

LSTM-based mental health chatbots represent a promising approach to providing accessible and personalized mental health support. By leveraging advanced NLP techniques, these chatbots can engage users in meaningful conversations, offer emotional support, and deliver evidence-based therapeutic interventions. As research and development in this field continue to advance, these tools have the potential to significantly improve mental health care accessibility and outcomes.

CHAPTER – 3

REQUIREMENT ANALYSIS

This project tackles the challenge of providing emotional support and motivation to students and individuals facing depression. Traditional support systems may not always be accessible or sufficient, necessitating an innovative solution that provides continuous, personalized emotional support and motivation. By leveraging technology, this project aims to empower users with the tools they need to improve their mental well-being and overcome their challenges.

2.1 Requirement Specifications:

2.1.1 Functional Requirements:

1. **Motivational Support and Response Generation:**
   - **Tailored Motivational Messages:**
     - The system shall generate tailored motivational messages based on user input.
     - This will be achieved using NLP techniques and LSTM-based RNN models to understand the user's emotional state and deliver relevant responses.
   - **Dynamic Response Generation:**
     - The chatbot will adjust its responses based on previous interactions and user behavior to ensure that the support provided is relevant and effective.
2. **Real-Time Emotional Support:**
   - **Instant Chat Functionality:**
     - The system shall integrate instant chat functionality to provide immediate emotional support.
     - The chatbot will be capable of responding to user queries and concerns in real time, offering comfort and guidance as needed.

   - **Daily Uplifting Messages:**
     - The system shall deliver daily uplifting affirmations and mental health tips to help users maintain a positive outlook and develop healthy coping strategies.

3. Interactive and User-Friendly Interface:

- **Seamless and Engaging Interface:**
  - The system shall provide a user-friendly and intuitive interface designed using Streamlit.
  - The interface will ensure a smooth and engaging user experience, allowing users to interact with the chatbot seamlessly.

- **Conversation Reset Feature:**
  - The system shall include a chat refresh option to allow users to clear previous conversations and start anew, ensuring a fresh and uncluttered interface.
- **Quick Response Prompts:**
  - The system shall provide quick response prompts for common questions that most users have asked.
  - Users can click on these prompts to receive immediate replies without having to type out their questions.

.

3. Performance Requirements:

- **Seamless Functionality:**
  - The system should function seamlessly across various devices with a user-friendly interface optimized for different screen sizes.
- **Response Time:**
  - The system response time for delivering content, processing user actions, and providing feedback should be within acceptable limits for a smooth support experience.

2.1.2 Non-Functional Requirements:

- **Speed and Efficiency:**
    - The chatbot should deliver content and feedback quickly to ensure a smooth user experience.
- **Scalability:**
    - The chatbot should function efficiently as the user base and content library grow..
- **Reliability:**
    - The chatbot should function consistently with minimal downtime

      avoid disruptions in support.
- **Usability:**
    - The chatbot should have a user-friendly interface that is intuitive for users to navigate.
- **Compatability:**
    - The chatbot should be compatible with various devices and platforms to ensure accessibility for all users.

2.1.3 Data Requirements:

- **Comprehensive Dataset for NLP Models:**
  - The system requires a comprehensive dataset comprising user conversations, therapeutic dialogues, and motivational messages.
  - This data will be used to train the NLP models for intent classification and response generation.
- **Evidence-Based Mental Health Support:**
  - Data on techniques and strategies for dealing with depression will be sourced from established books in the field of psychology and mental health.This information will be integrated into the chatbot to provide evidence-based support and guidance.

2.2 Workflow Environment :

This project aims to deliver personalized motivation and emotional support to students and individuals experiencing depression through a chatbot powered by NLP and LSTM-based RNN models. Below is an overview of the workflow environment:

1. Data Collection:

1.1 Data Collection Methods:

- Collect comprehensive datasets including user conversations, therapeutic dialogues, and motivational messages.
- Source evidence-based techniques and strategies for mental health support from reputable psychology and mental health books.

1.2 Data Analysis and Insights:

- **Quantitative Analysis:** Analyzed structured data to identify patterns in user needs and preferences.
- **Qualitative Analysis:** Used thematic analysis to interpret data from books and datasets, providing deeper insights into users' emotional states and motivational needs.
- **Key Findings:** Identified critical areas where users require the most support, such as motivational messages, cop` ing strategies, and real-time assistance.

.

2. Design Phase:

2.1 Understanding User Needs:

- **User Research:** Conducted research to understand user needs and preferences, informing the design of the chatbot interface.
- **Personas and User Journeys:** Developed detailed personas and user journey maps to ensure the design addressed specific user needs and contexts.

2.2 Interface and Model Design:

- **Initial Structuring:** Created initial wireframes to outline the basic structure and layout of the chatbot interface.
- **High-Fidelity Prototypes:** Developed high-fidelity prototypes incorporating user feedback from initial tests.
- **Design Refinement:** Conducted multiple iterations based on user feedback to refine the design, ensuring a balance between aesthetics and functionality.
- **UI Layout Planning:** Designed the user interface, determining the placement of the search bar, chat history, most-asked questions, and the overall look.
- **LSTM Framework Development:** Designed the LSTM model architecture, including adding layers, activation functions, and dropout layers.

3. Development Phase:

3.1 Tech Stack Learning:

- **Training Programs:** Enrolled in training programs to master ANN, LSTM- based RNN, and NLP techniques.
- **Practice Projects:** Completed practice projects to gain hands-on experience and confidence in using these technologies.

3.2 Chatbot Development:

- **Frontend Development:**
    - **Translating Designs to Code:** Converted detailed design prototypes into functional code using Streamlit, ensuring every UI element was accurately represented and fully operational.
    - **Responsive Design:** Implemented a responsive design to ensure the chatbot works seamlessly across different devices and screen sizes.
    - **Interactive Elements:** Incorporated features to enhance the user experience, making the chatbot engaging and user-friendly.
    - **Testing and Quality Assurance:** Conducted rigorous testing to identify and fix bugs, ensuring high standards of quality and functionality.

3.3 NLP and RNN Integration:

- **NLP and RNN Integration:** Integrated NLP techniques and LSTM-based RNN models into the chatbot to handle a wide range of user queries and provide relevant responses.
- **User Testing:** Conducted user testing to ensure the chatbot was effective and user-friendly, making necessary adjustments based on feedback.

4. Implementing Key Features:

4.1 Personalized Support:

- **Personalized Motivational Messages:** Developed a system to tailor motivational messages to each user's emotional state and needs.
- **Adaptive Response Generation:** Continuously adjusted response generation based on user interactions to maintain relevance and effectiveness.

4.2 Real-Time Emotional Support:

- **Real-Time Chat Functionality:** Implemented real-time chat functionality to provide immediate support.
- **Daily Positive Affirmations:** Delivered daily positive affirmations and mental health tips to help users maintain a positive outlook.

4.3 Interactive Interface:

- **Chat Refresh Option:** Included a chat refresh option to clear previous conversations and start anew.
- **Previous Prompts for Similar Questions:** Provided previous prompts for similar questions, allowing users to click on these prompts to receive immediate replies without typing out their questions.

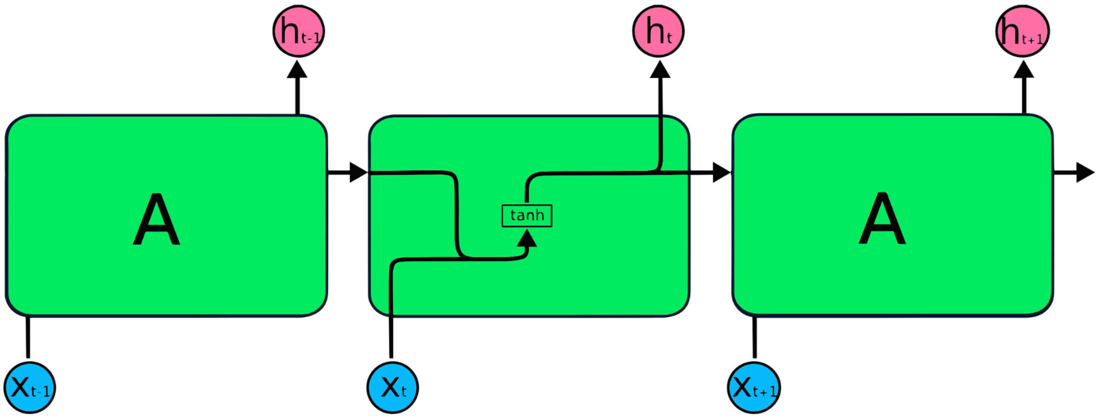
CHAPTER – 4

CHATBOT ARCHITECTURE

4.1 Long Short-Term Memory (LSTM)

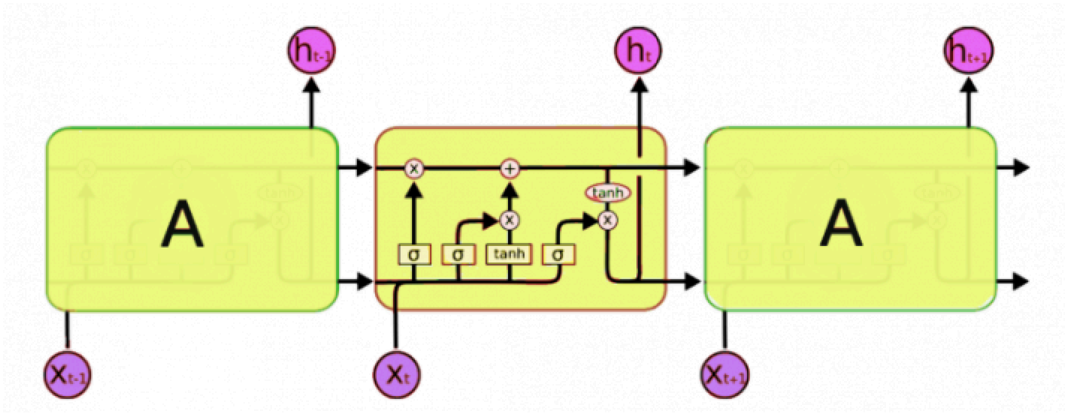
LSTMs come to the rescue to solve the vanishing gradient problem. It does so by ignoring (forgetting) useless data/information in the network. The LSTM will forget the data if there is no useful information from other inputs (prior sentence words). When new information comes, the network determines which information to be overlooked and which to be remembered.

LSTM Architecture

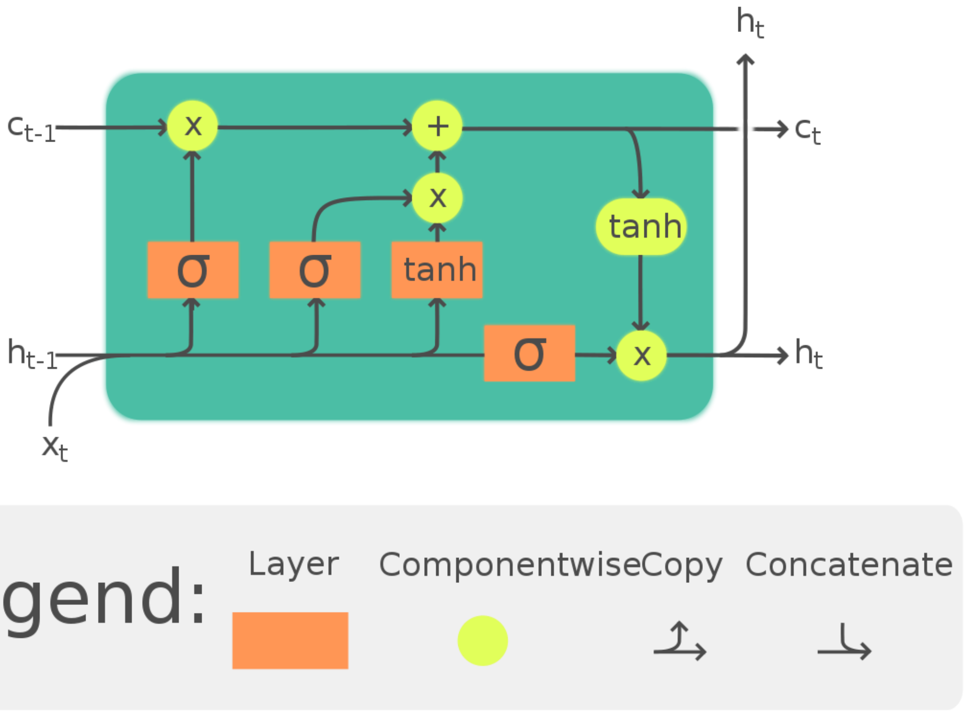


RNN STRUCTURE

In LSTMs, instead of just a simple network with a single activation function, we have multiple components, giving power to the network to forget and remember information.



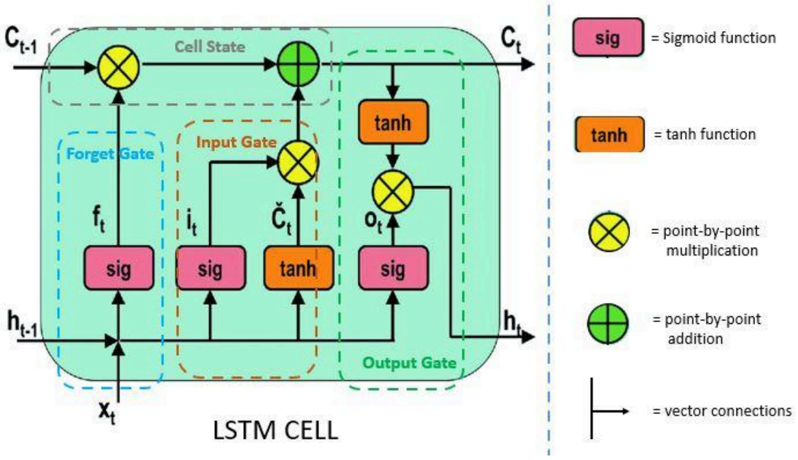
LSTM ARCHITECTURE



NOTATIONS

LSTMs have 4 different components, namely

1. Cell state (Memory cell)
2. Forget gate
3. Input gate
4. Output gate

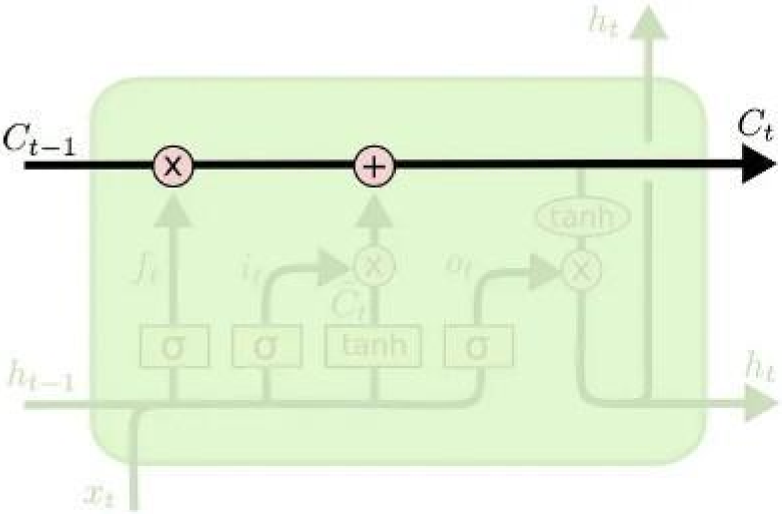


LSTM CELL

LSTMS COMPONENTS

1.Cell State (Memory cell)

- The **Cell State** is the first component of the LSTM that runs through the entire LSTM unit. It can be thought of as a conveyor belt that carries information across the sequence without significant changes. This state is regulated by gates which add or remove information, allowing the cell state to maintain important data over long sequences.

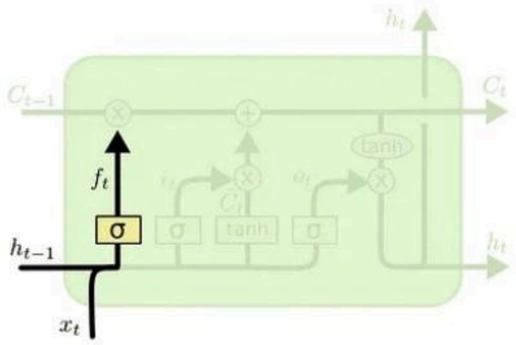


LSTM CELL STATE

- **Pointwise Operation (X)**: This involves multiplying the cell state by an array of [-1, 0, 1], where:
    - Multiplying by -1 implies forgetting (removing) some information from the cell state.
    - Multiplying by 0 means the information remains unchanged (remembering).
    - Multiplying by 1 involves adding new information to the cell state.

2. Forget Gate

The forget LSTM gate, as the name suggests, decides what information should be forgotten. A sigmoid layer is used to make this decision. This sigmoid layer is called the “forget gate layer”.



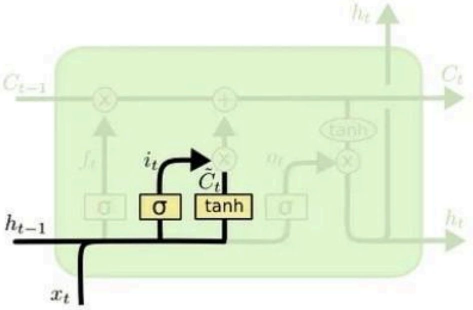
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

LSTM FORGET GATE

It does a dot product of *h(t-1)* and *x(t)* and with the help of the sigmoid layer, outputs a number between 0 and 1 for each number in the cell state *C(t-1)*. If the output is a ‘1’, it means we will keep it. A ‘0’ means to forget it completely.

3. Input gate

The input gate gives new information to the LSTM and decides if that new information is going to be stored in the cell state.

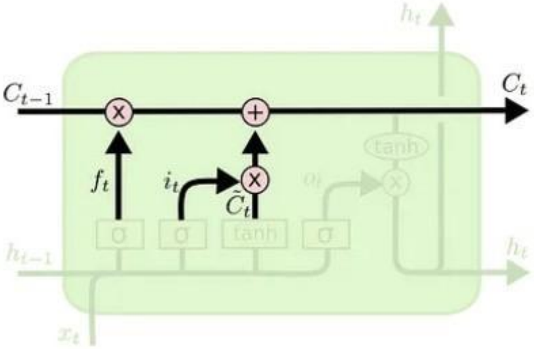


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

This has 3 parts-

1.A *sigmoid* layer decides the values to be updated. This layer is called the “input gate layer”
2.A *tanh* activation function layer creates a vector of new candidate values, *Č(t)*, that could be added to the state.
3.Then we combine these 2 outputs, *i(t)* * *Č(t),* and update the cell state.

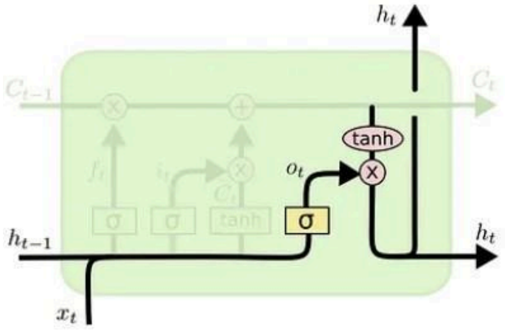
The new cell state *C(t)* is obtained by adding the output from forget and input gates.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

4. Output gate

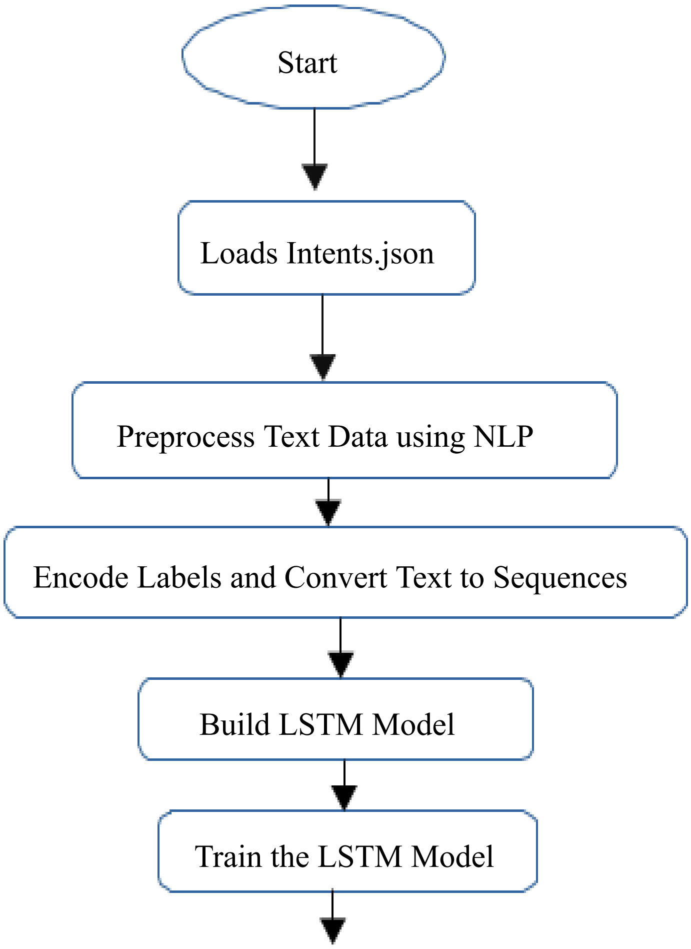
The output of the LSTM unit depends on the new cell state.



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

First, a sigmoid layer decides what parts of the cell state we're going to output. Then, a *tanh* layer is used on the cell state to squash the values between -1 and 1, which is finally multiplied by the sigmoid gate output.

4.2 CHATBOT WORK FLOW



Start

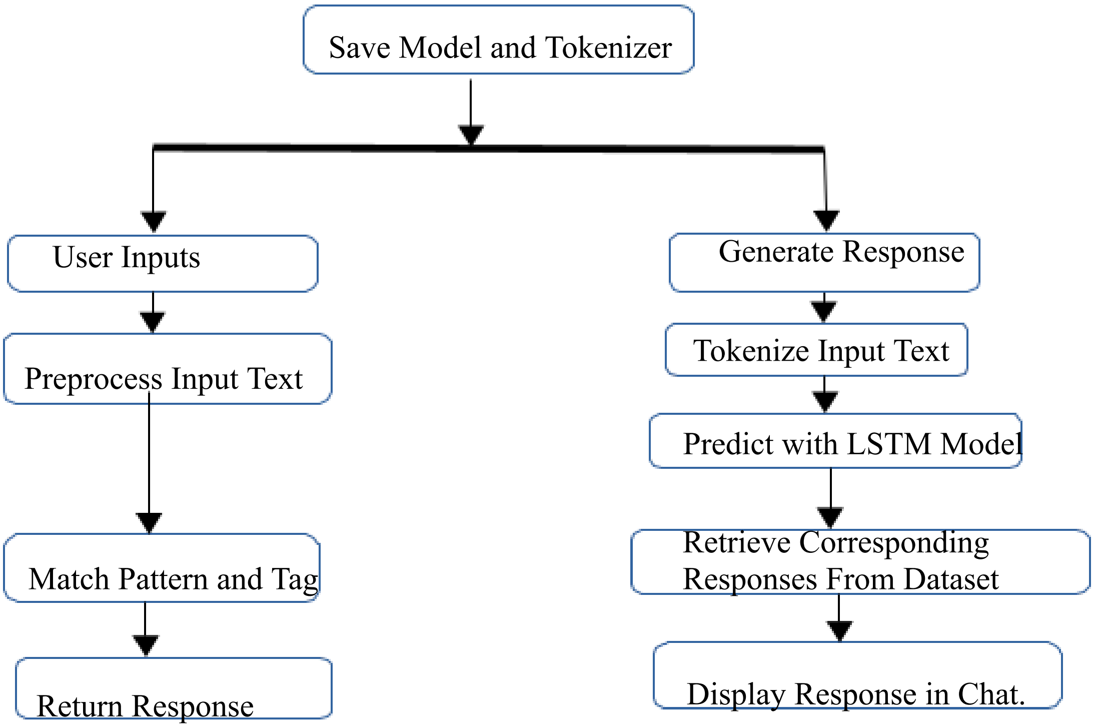
Loads Intents.json

Preprocess Text Data using NLP

Encode Labels and Convert Text to Sequences

Build LSTM Model

Train the LSTM Model

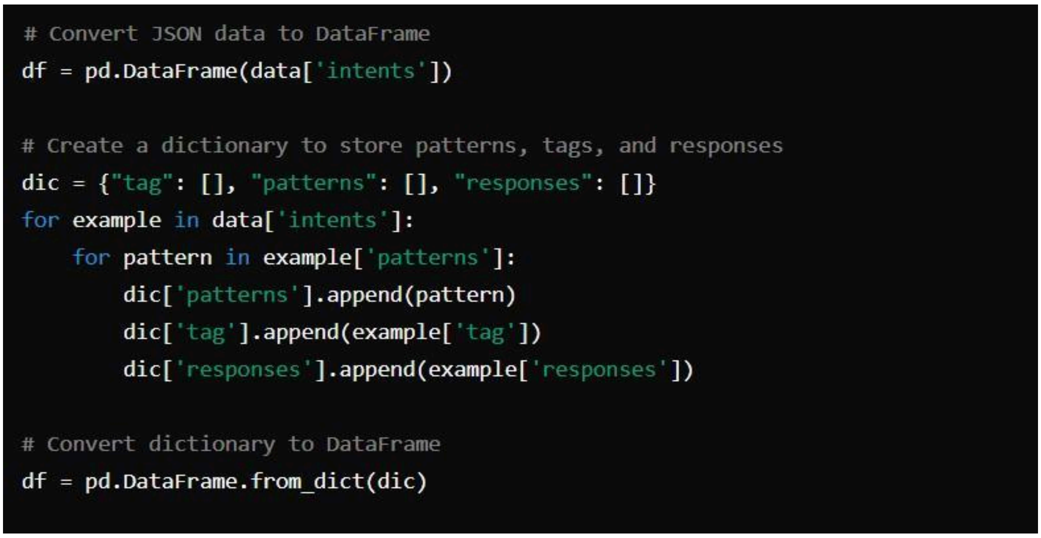


1. Start

The process begins here.

2. Load Intents.json

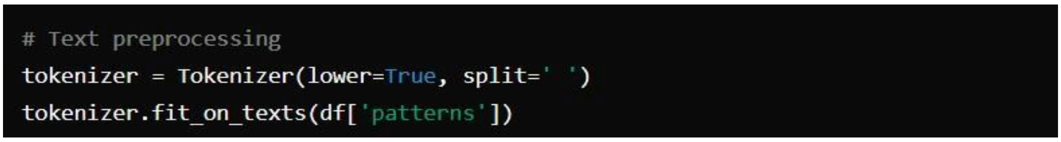
- **Description**: Load the Intents.json file which contains predefined patterns and responses.
- **Details**:
    - This JSON file typically contains various intents (user queries) and corresponding tags, patterns, and responses.



```
# Convert JSON data to DataFrame
df = pd.DataFrame(data['intents'])

# Create a dictionary to store patterns, tags, and responses
dic = {"tag": [], "patterns": [], "responses": []}
for example in data['intents']:
    for pattern in example['patterns']:
        dic['patterns'].append(pattern)
        dic['tag'].append(example['tag'])
        dic['responses'].append(example['responses'])

# Convert dictionary to DataFrame
df = pd.DataFrame.from_dict(dic)
```

3. Preprocess Text Data using NLP

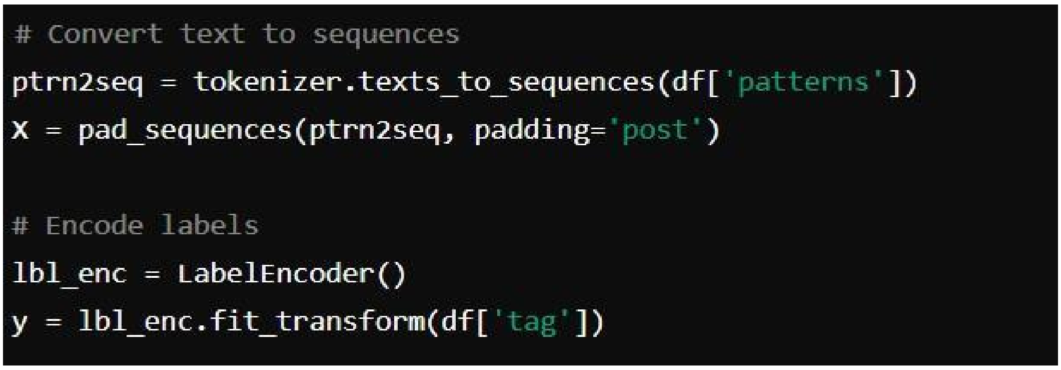
- **Description**: Use Natural Language Processing (NLP) techniques to clean and preprocess the text data.
- **Details**:
  - Tokenization: Splitting text into words or tokens.
  - Lowercasing: Converting text to lowercase to maintain uniformity.



```
# Text preprocessing
tokenizer = Tokenizer(lower=True, split=' ')
tokenizer.fit_on_texts(df['patterns'])
```

4. Encode Labels and Convert Text to Sequences

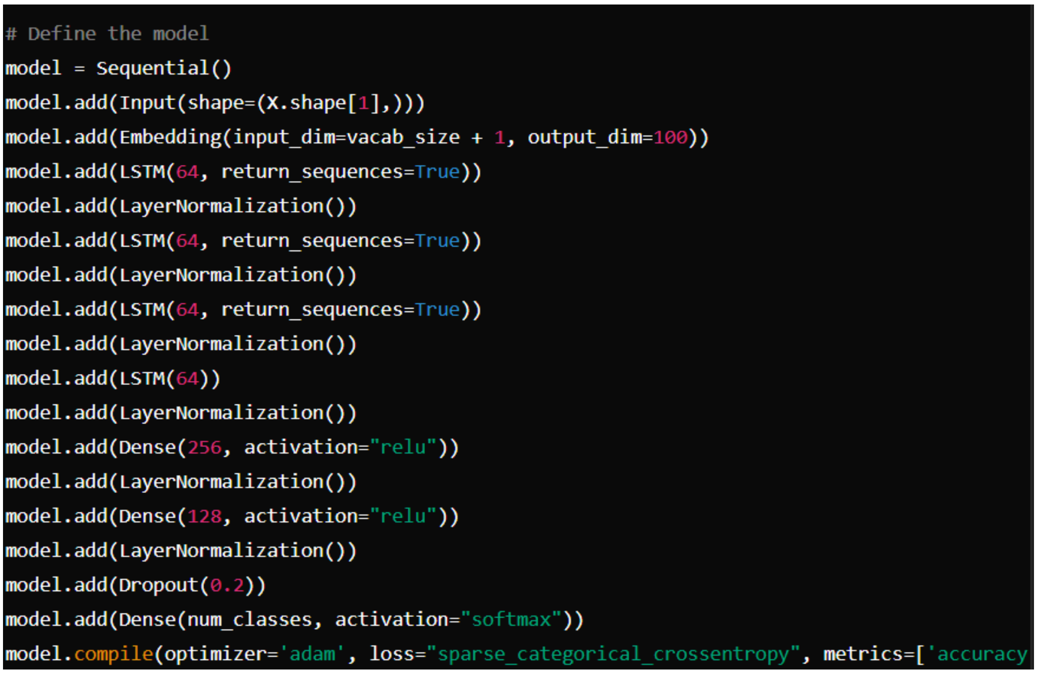
- **Description**: Convert the preprocessed text into numerical form for model input.
- **Details**:
  - **Label Encoding**: Assign numerical values to categorical labels (e.g., "greeting" -> 0, "goodbye" -> 1).
  - **Text to Sequences**: Convert text data into sequences of integers using a tokenizer. Each word/token is replaced by its index in a vocabulary.



```
# Convert text to sequences
ptrn2seq = tokenizer.texts_to_sequences(df['patterns'])
X = pad_sequences(ptrn2seq, padding='post')

# Encode labels
lbl_enc = LabelEncoder()
y = lbl_enc.fit_transform(df['tag'])
```

5. Build LSTM Model

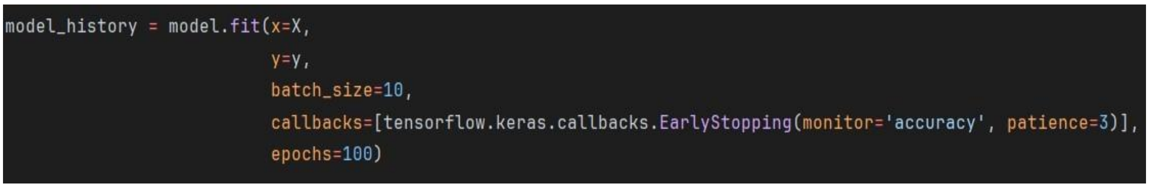
- **Description**: Define the architecture of the Long Short-Term Memory (LSTM) model.
- **Details**:
  - Input Layer: Define the input shape.
  - Embedding Layer: Convert integer sequences into dense vectors of fixed size.
  - LSTM Layer(s): Add one or more LSTM layers to capture temporal dependencies.
  - Dense Layer: Add fully connected layers for output.
  - Output Layer: Use a softmax activation function for classification tasks.



```
# Define the model
model = Sequential()
model.add(Input(shape=(X.shape[1],)))
model.add(Embedding(input_dim=vacab_size + 1, output_dim=100))
model.add(LSTM(64, return_sequences=True))
model.add(LayerNormalization())
model.add(LSTM(64, return_sequences=True))
model.add(LayerNormalization())
model.add(LSTM(64, return_sequences=True))
model.add(LayerNormalization())
model.add(LSTM(64))
model.add(LayerNormalization())
model.add(Dense(256, activation="relu"))
model.add(LayerNormalization())
model.add(Dense(128, activation="relu"))
model.add(LayerNormalization())
model.add(Dropout(0.2))
model.add(Dense(num_classes, activation="softmax"))
model.compile(optimizer='adam', loss="sparse_categorical_crossentropy", metrics=['accuracy
```

6. Train the LSTM Model

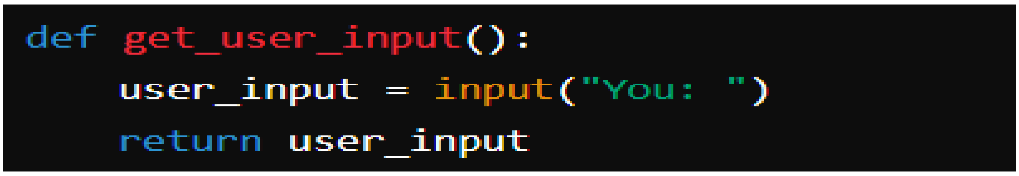
- **Description**: Train the LSTM model using the preprocessed data.
- **Details**:
    - Compile the model with appropriate loss function, optimizer, and metrics.
    - Fit the model on the training data with specified epochs and batch size.
    - Monitor training and validation loss/accuracy.



```
model_history = model.fit(x=X,
                          y=y,
                          batch_size=10,
                          callbacks=[tensorflow.keras.callbacks.EarlyStopping(monitor='accuracy', patience=3)],
                          epochs=100)
```

7. User Inputs

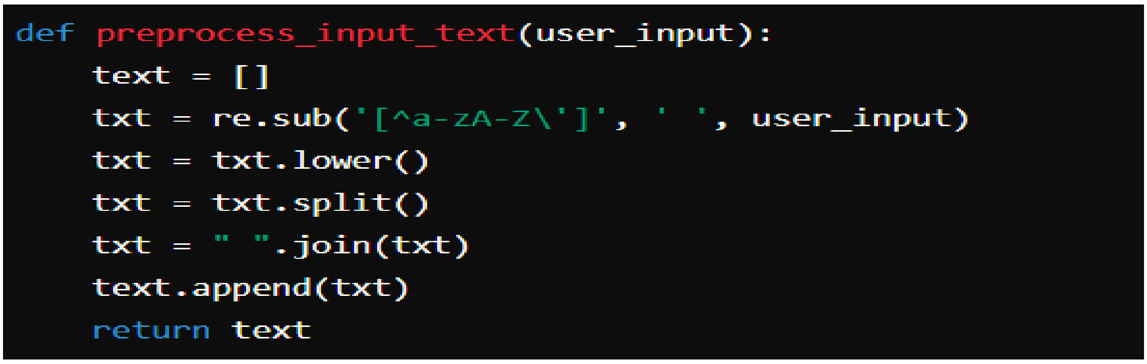
- **Description**: Accept user input from the chat interface.
- **Details**:
    - Capture the user's query or message.



```
def get_user_input():
    user_input = input("You: ")
    return user_input
```

8. Preprocess Input Text

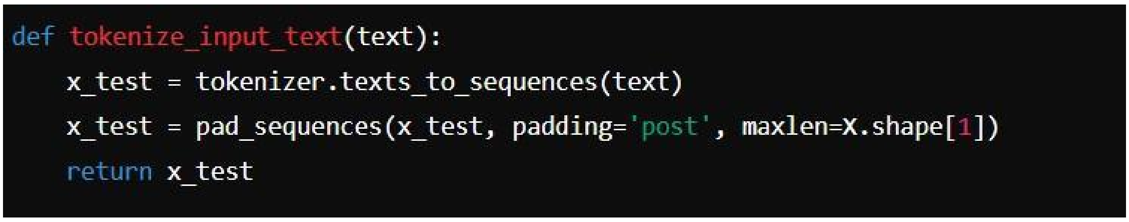
- **Description**: Apply the same preprocessing steps to the user input as done during training.
- **Details**:
    - Tokenization, lowercasing, removing punctuation and stop words, lemmatization/stemming.



```
def preprocess_input_text(user_input):
    text = []
    txt = re.sub('[^a-zA-Z\']', ' ', user_input)
    txt = txt.lower()
    txt = txt.split()
    txt = " ".join(txt)
    text.append(txt)
    return text
```

9. Tokenize Input Text

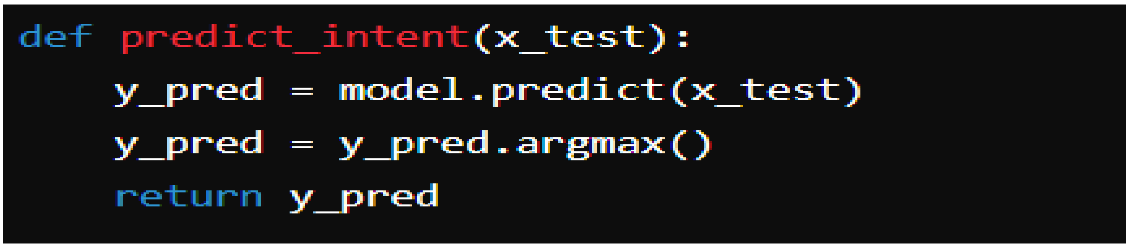
- **Description**: Convert the preprocessed user input text into numerical sequences using the saved tokenizer.
- **Details**:
    - Transform the text into sequences of integers.



```
def tokenize_input_text(text):
    x_test = tokenizer.texts_to_sequences(text)
    x_test = pad_sequences(x_test, padding='post', maxlen=X.shape[1])
    return x_test
```

10. Predict with LSTM Model

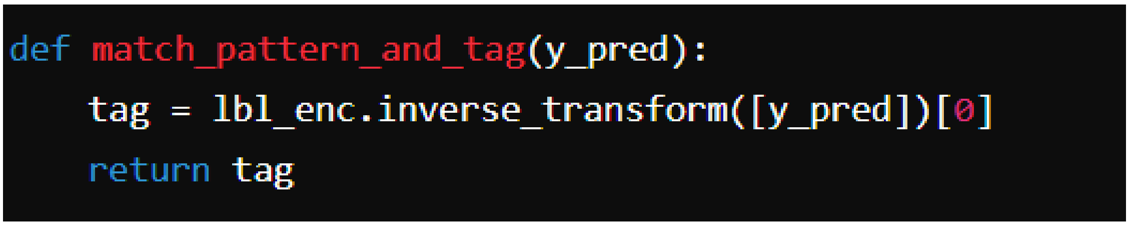
- **Description**: Use the trained LSTM model to predict the intent of the user input.
- **Details:**Pass the tokenized input sequence to the model.



```
def predict_intent(x_test):
    y_pred = model.predict(x_test)
    y_pred = y_pred.argmax()
    return y_pred
```

11. Retrieve Corresponding Responses from Dataset

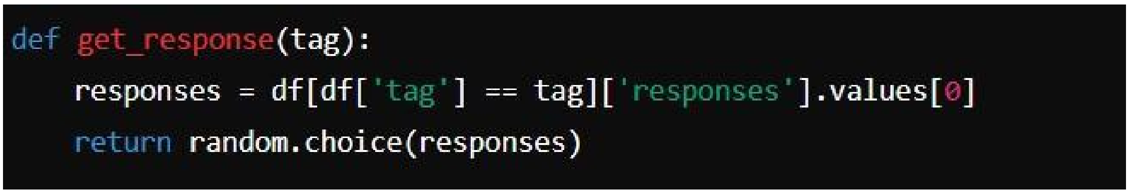
- **Description**: Match the predicted intent with predefined patterns and tags in the Intents.json file.
- **Details:**Compare the model's predicted intent with the intents in the JSON file.



```
def match_pattern_and_tag(y_pred):
    tag = lbl_enc.inverse_transform([y_pred])[0]
    return tag
```

12. Retrieve Corresponding Responses from Dataset

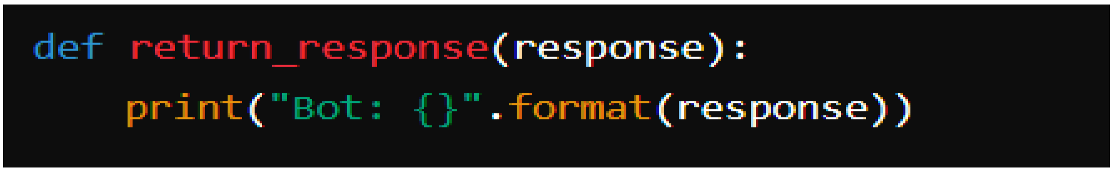
- **Description**: Fetch the appropriate response from the dataset based on the matched intent.
- **Details**:
    - Select a response from the list of predefined responses for the matched intent.



```python
def get_response(tag):
    responses = df[df['tag'] == tag]['responses'].values[0]
    return random.choice(responses)
```

13. Return Response

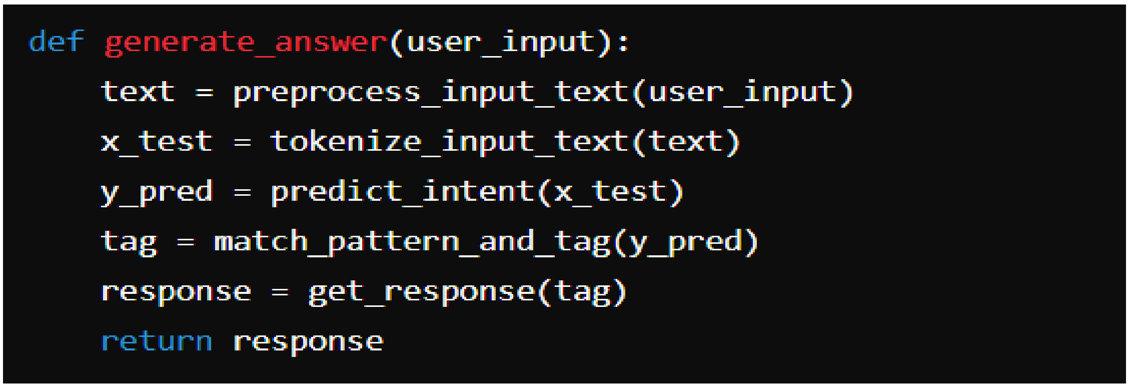
- **Description**: Generate and return the response to the user.
- **Details**: Send the response back to the chat interface.



```python
def return_response(response):
    print("Bot: {}".format(response))
```

14. Generate Response

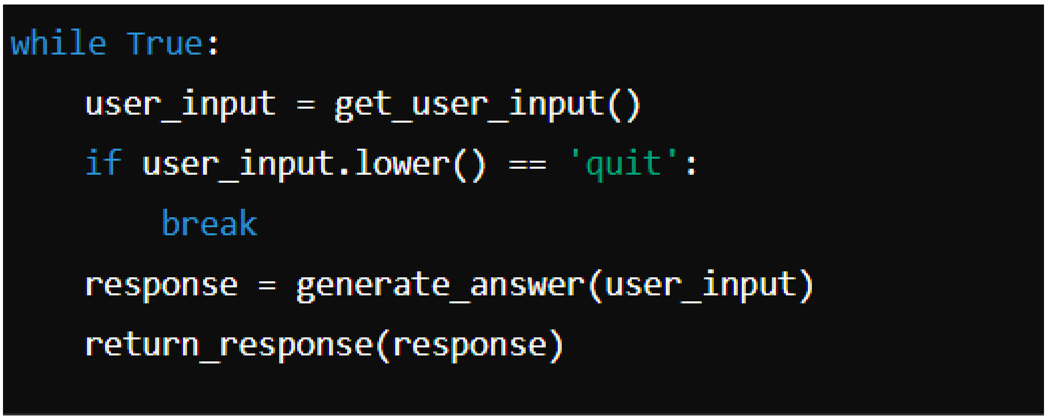
- **Description**: Generate the final response based on the user input and model prediction.
- **Details**:
    - Process the response to ensure it is coherent and contextually relevant.



```python
def generate_answer(user_input):
    text = preprocess_input_text(user_input)
    x_test = tokenize_input_text(text)
    y_pred = predict_intent(x_test)
    tag = match_pattern_and_tag(y_pred)
    response = get_response(tag)
    return response
```

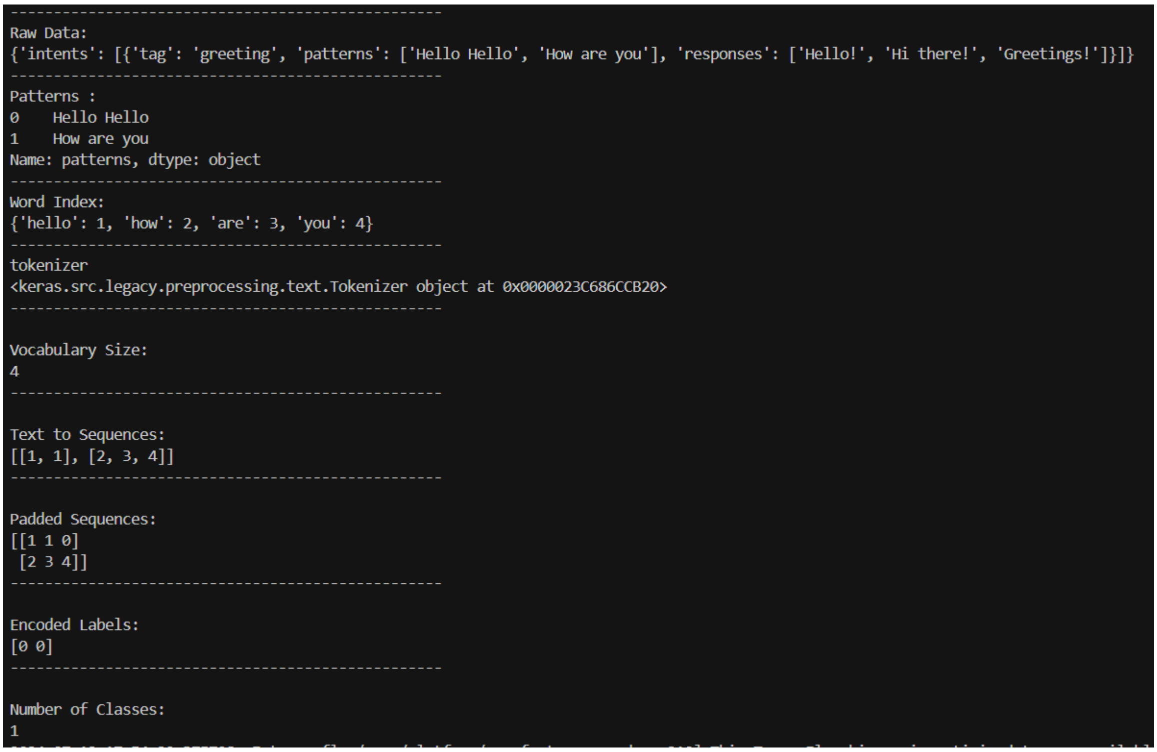
15. Display Response in Chat

- **Description**: Display the generated response in the chat interface.
- **Details**:
  - Update the chat interface with the bot's response



```
while True:
    user_input = get_user_input()
    if user_input.lower() == 'quit':
        break
    response = generate_answer(user_input)
    return_response(response)
```

Example : Workflow of Data Preprocessing



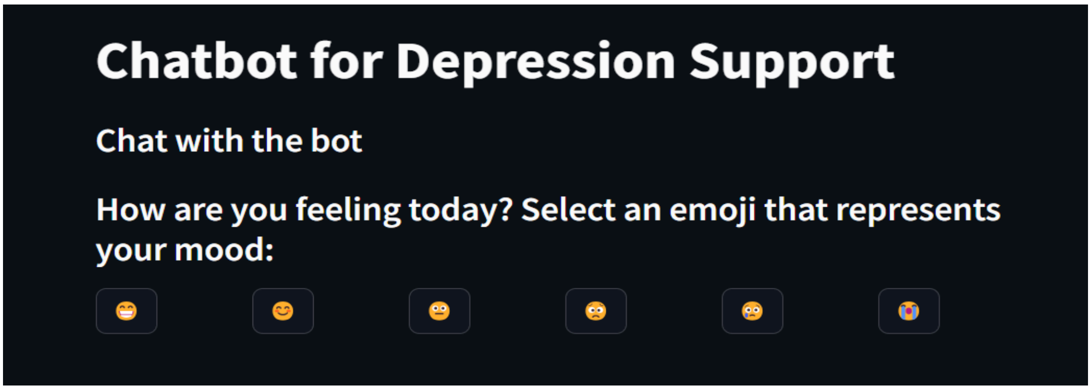
```
--------------------------------------------------
Raw Data:
{'intents': [{'tag': 'greeting', 'patterns': ['Hello Hello', 'How are you'], 'responses': ['Hello!', 'Hi there!', 'Greetings!']}]}
--------------------------------------------------
Patterns :
0    Hello Hello
1    How are you
Name: patterns, dtype: object
--------------------------------------------------
Word Index:
{'hello': 1, 'how': 2, 'are': 3, 'you': 4}
--------------------------------------------------
tokenizer
<keras.src.legacy.preprocessing.text.Tokenizer object at 0x0000023C686CCB20>
--------------------------------------------------

Vocabulary Size:
4
--------------------------------------------------

Text to Sequences:
[[1, 1], [2, 3, 4]]
--------------------------------------------------

Padded Sequences:
[[1 1 0]
 [2 3 4]]
--------------------------------------------------

Encoded Labels:
[0 0]
--------------------------------------------------

Number of Classes:
1
```

CHAPTER - 5

CHATBOT INTERFACE

1. Input Prompt Box Feature

1. **User Emotion Emoji:**

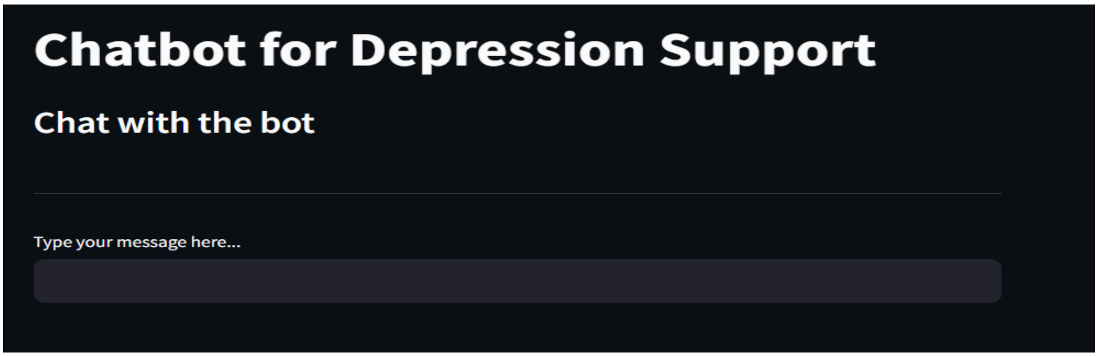


Chatbot for Depression Support

Chat with the bot

How are you feeling today? Select an emoji that represents your mood:

2. **User Input Box:**

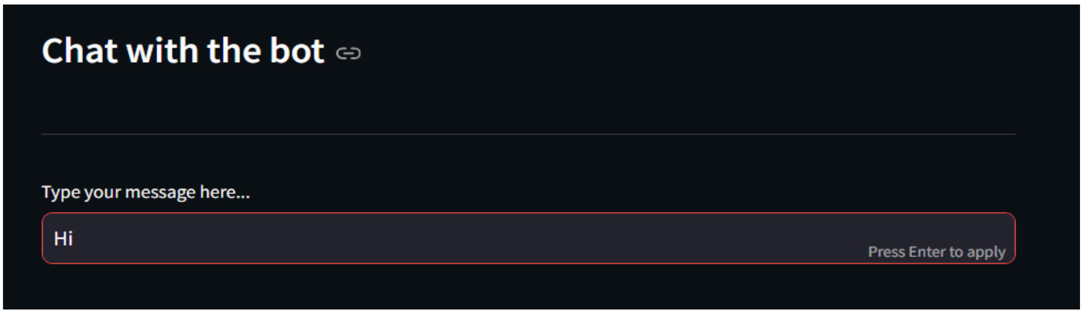


Chatbot for Depression Support

Chat with the bot

Type your message here...

3. **Typing a Message**:

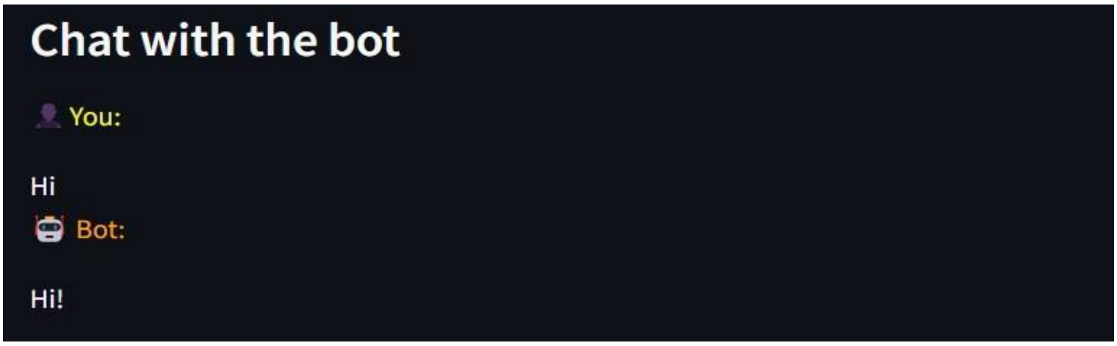


Chat with the bot

Type your message here...

Hi

4. **Message Sent**:



Chat with the bot

👤 You:

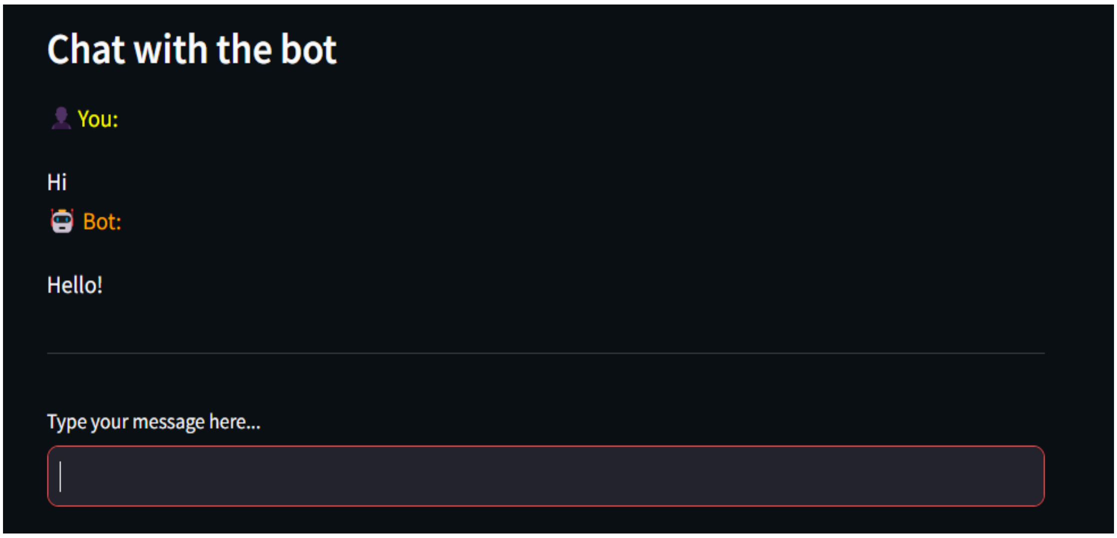
Hi

🤖 Bot:

Hi!

2.Refresh Chat Feature

1.Initial Chat Interface:



Chat with the bot

👤 You:

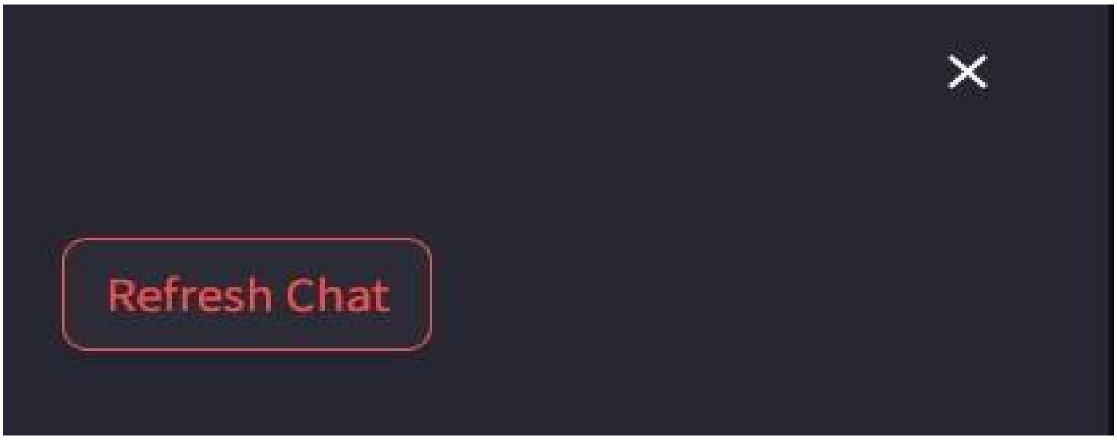
Hi

🤖 Bot:

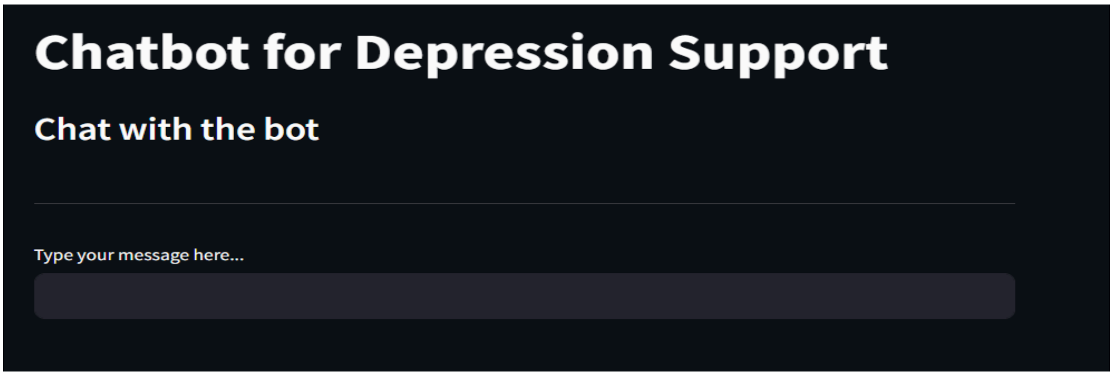
Hello!

Type your message here...

2.Click on "Refresh Chat" Button:



3.Chat Refreshed :

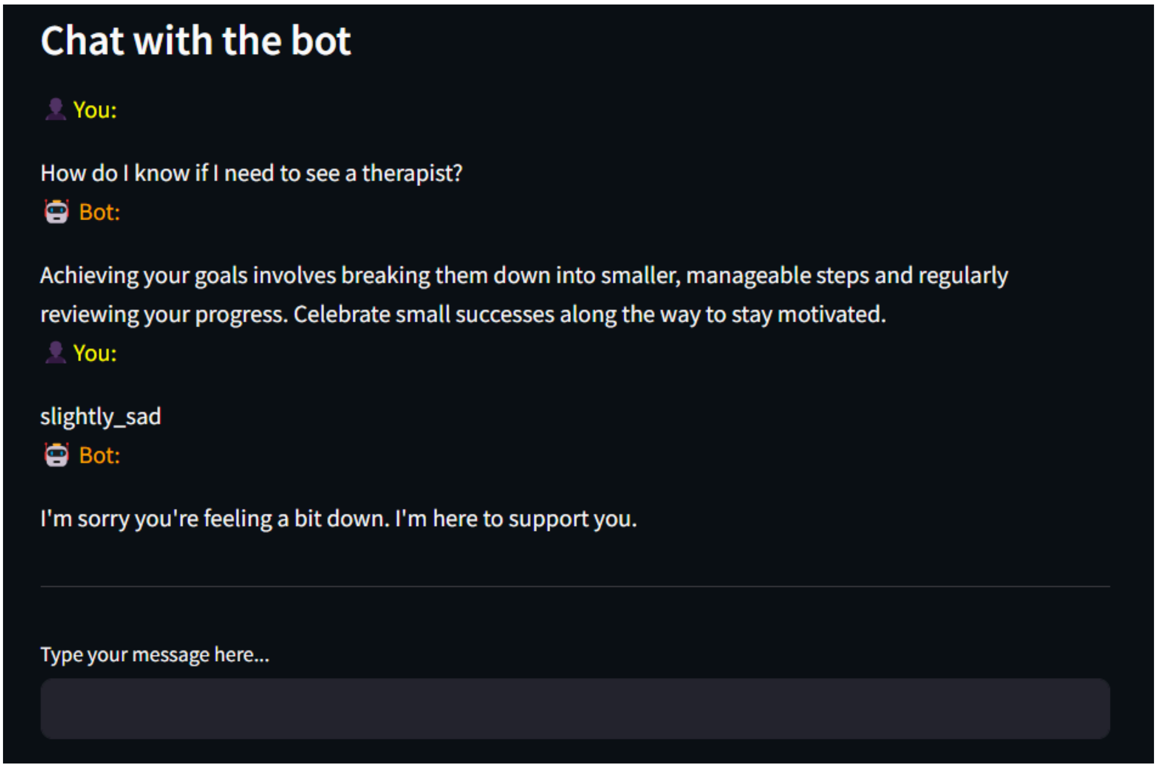


Chatbot for Depression Support

Chat with the bot

3.Clear Chat History Feature

1.Start a new Conversation :



Chat with the bot

👤 You:

How do I know if I need to see a therapist?

🤖 Bot:

Achieving your goals involves breaking them down into smaller, manageable steps and regularly reviewing your progress. Celebrate small successes along the way to stay motivated.

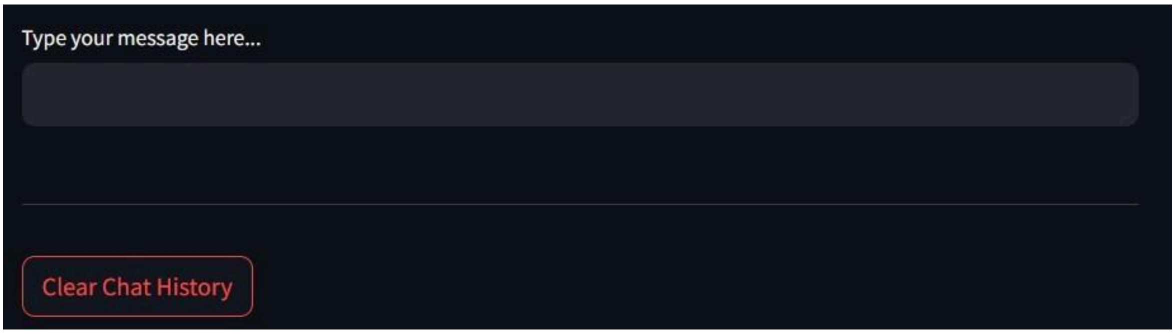
👤 You:

slightly_sad

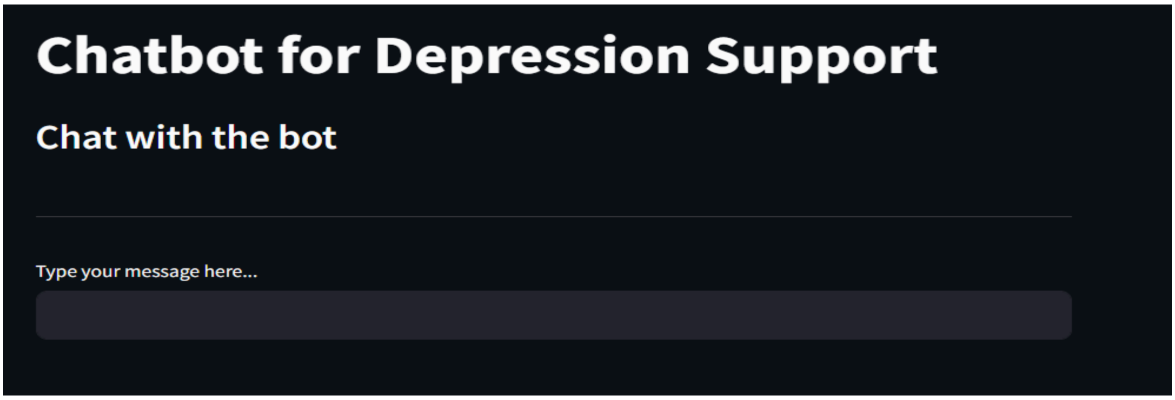
🤖 Bot:

I'm sorry you're feeling a bit down. I'm here to support you.

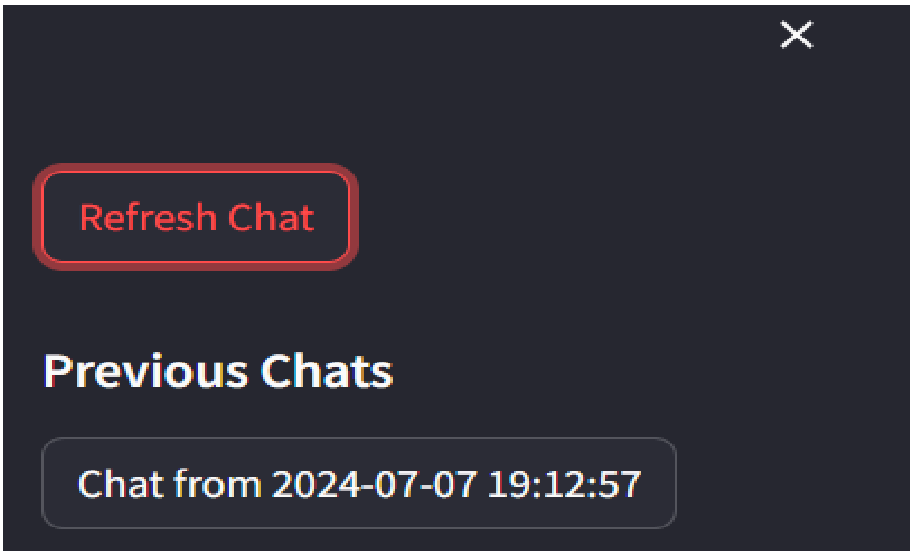
2.Click on "Clear Chat History" Button:



3.Chat History Cleared:

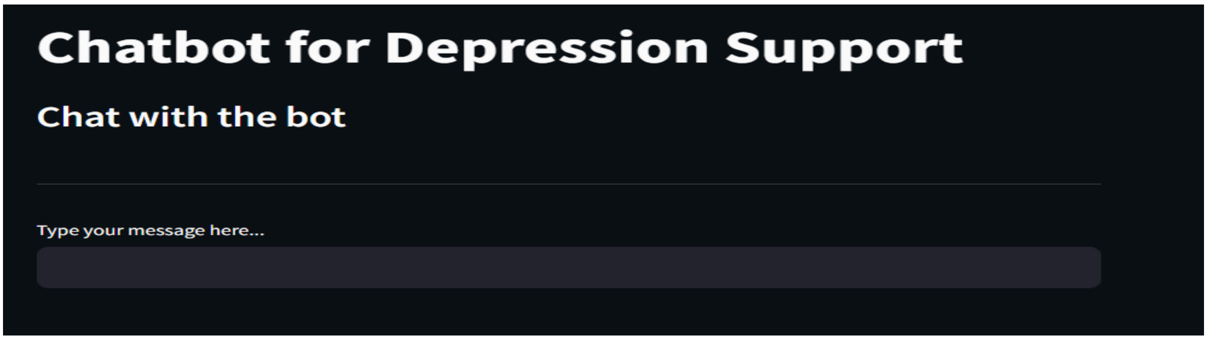


4.Chat History Stored :



4. Input Prompt Box Feature

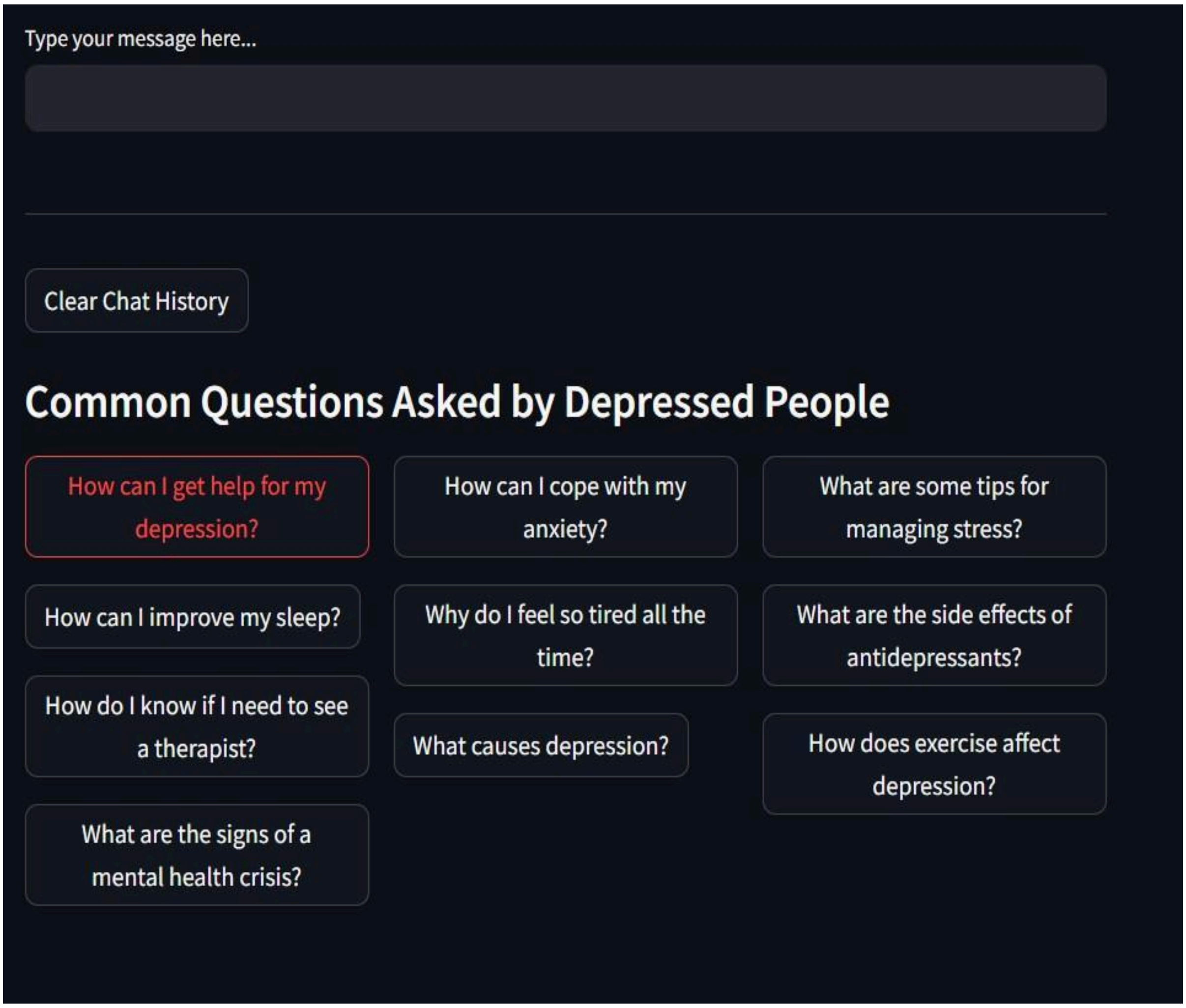
1.User Input Box:



Chatbot for Depression Support

Chat with the bot

2.Selecting a Message:

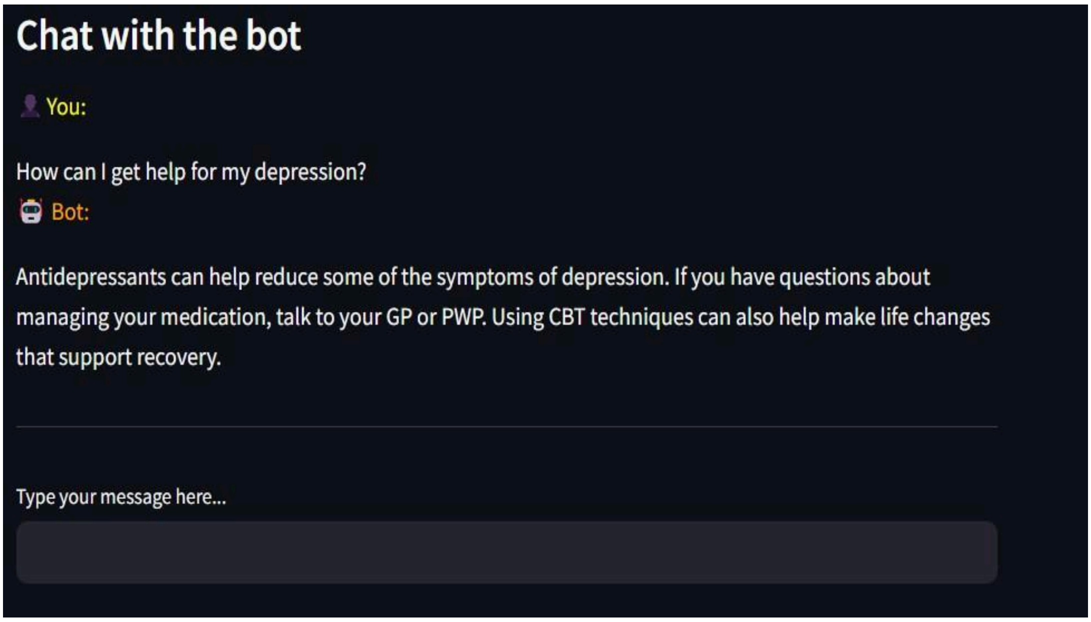


Type your message here...
Clear Chat History

Common Questions Asked by Depressed People

How can I get help for my depression?
How can I cope with my anxiety?
What are some tips for managing stress?
How can I improve my sleep?
Why do I feel so tired all the time?
What are the side effects of antidepressants?
How do I know if I need to see a therapist?
What causes depression?
How does exercise affect depression?
What are the signs of a mental health crisis?

3.Message sent :



Chat with the bot

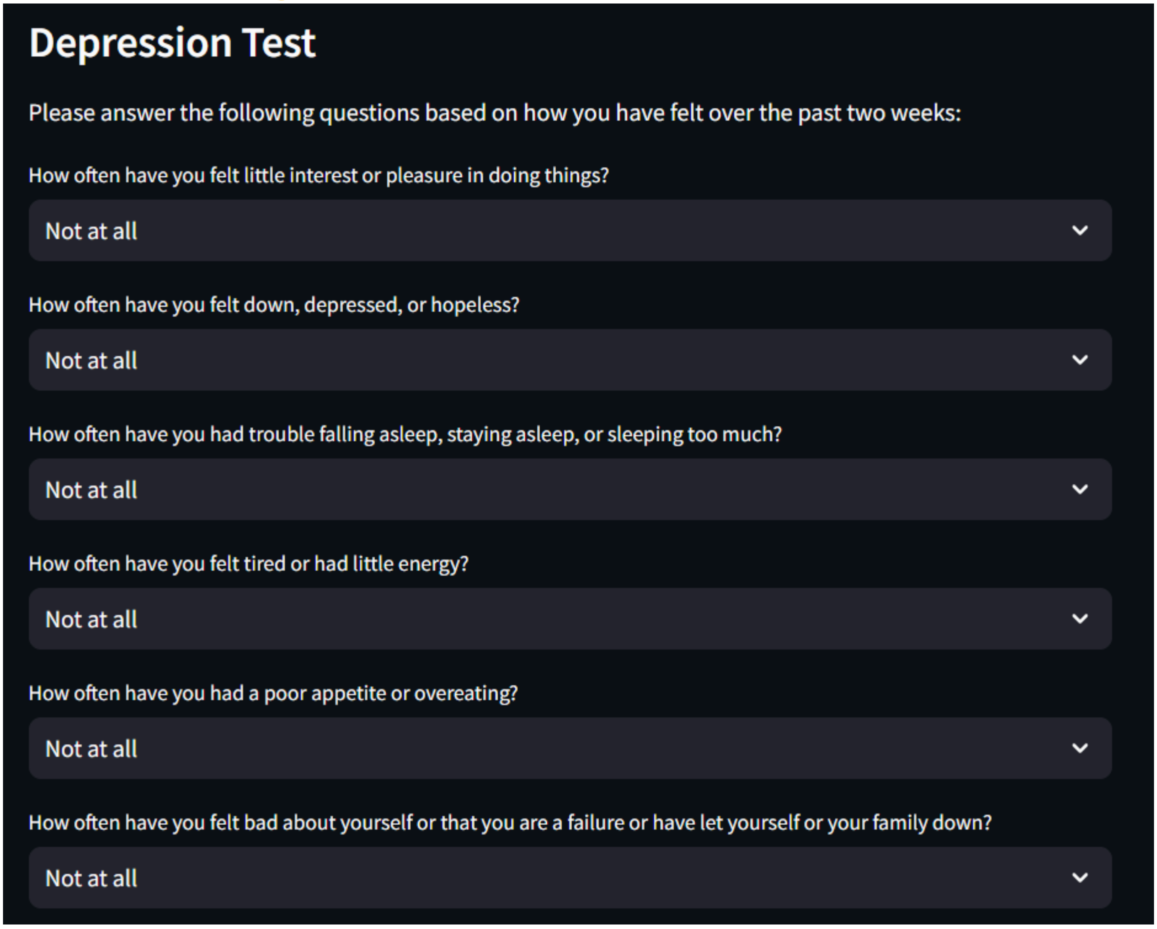
👤 You:

How can I get help for my depression?

🤖 Bot:

Antidepressants can help reduce some of the symptoms of depression. If you have questions about managing your medication, talk to your GP or PWP. Using CBT techniques can also help make life changes that support recovery.

4. Depression Test 👍



Depression Test

Please answer the following questions based on how you have felt over the past two weeks:

How often have you felt little interest or pleasure in doing things?

Not at all

How often have you felt down, depressed, or hopeless?

Not at all

How often have you had trouble falling asleep, staying asleep, or sleeping too much?

Not at all

How often have you felt tired or had little energy?

Not at all

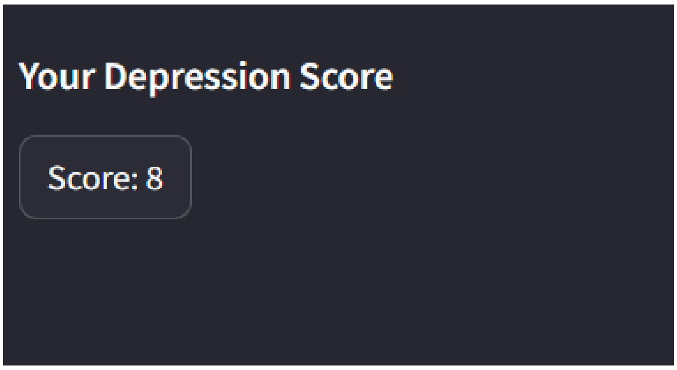
How often have you had a poor appetite or overeating?

Not at all

How often have you felt bad about yourself or that you are a failure or have let yourself or your family down?

Not at all

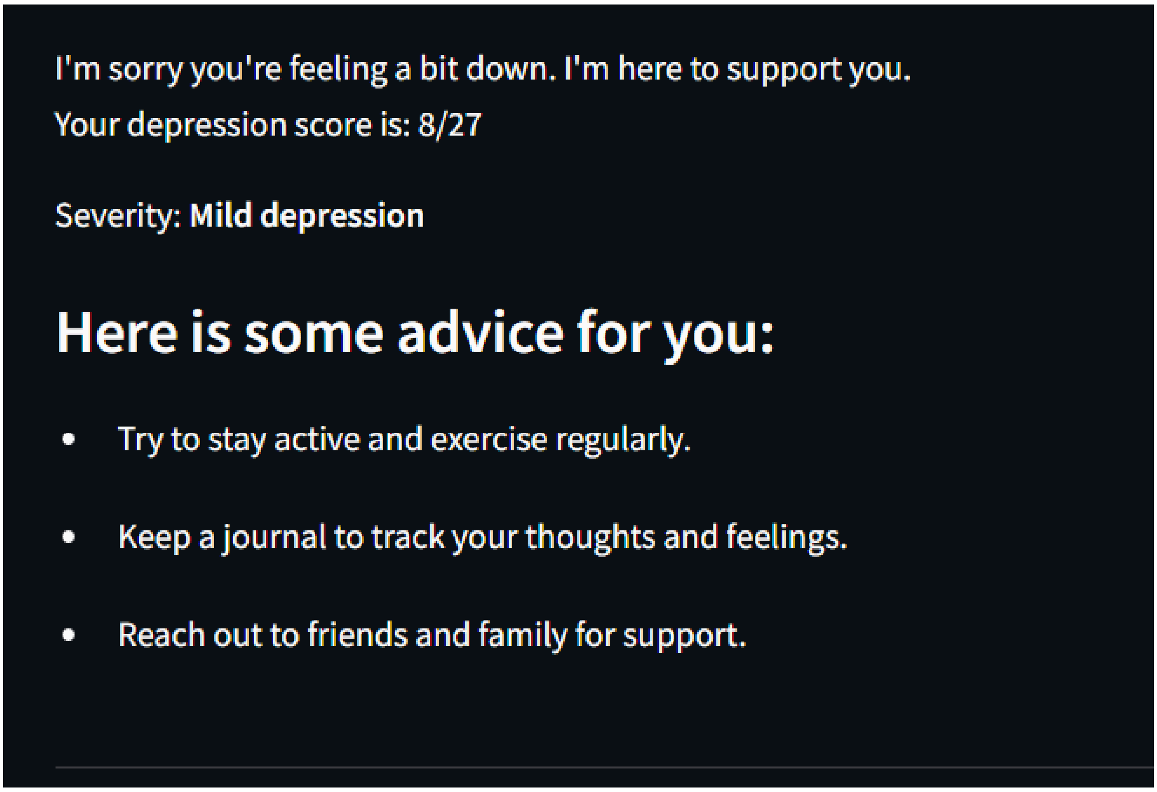
5. Depression Test Score :



Your Depression Score

Score: 8

6. Suggestions :



I'm sorry you're feeling a bit down. I'm here to support you.

Your depression score is: 8/27

Severity: **Mild depression**

Here is some advice for you:

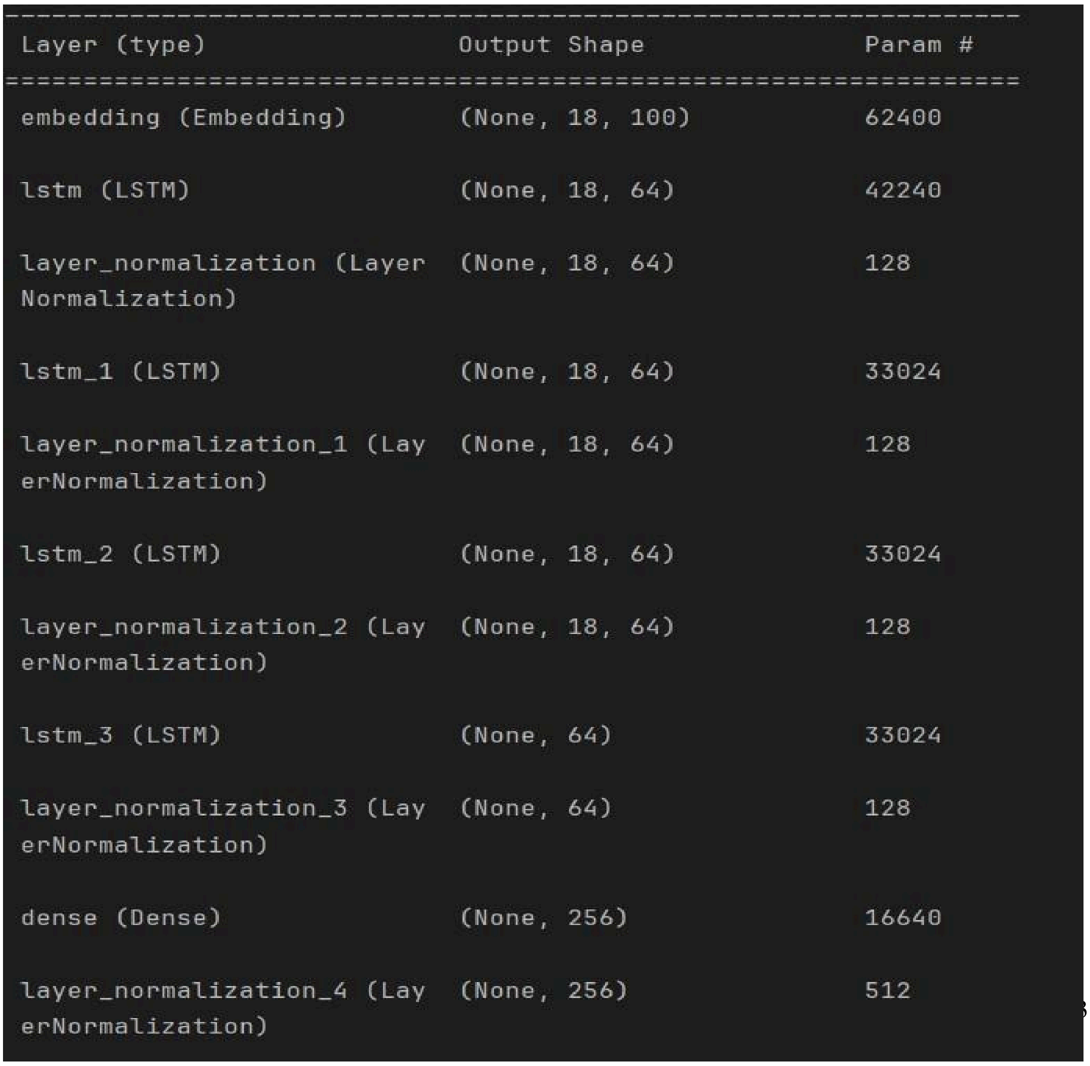
- Try to stay active and exercise regularly.
- Keep a journal to track your thoughts and feelings.
- Reach out to friends and family for support.

CHAPTER - 6

RESULTS AND DISCUSSION

6.1 Results

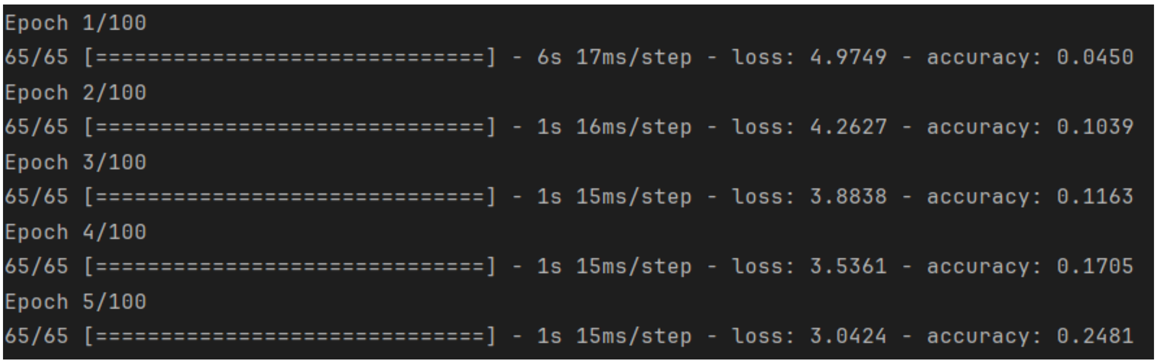
1. **Vocabulary Size**:
   - The vocabulary size determined by the tokenizer is printed as output, indicating the number of unique words identified from the patterns in the intents dataset.
   - Example Output: `1234` (This number will vary based on the actual dataset).
2. **Model Training**:
   - The model summary provides a detailed structure of the LSTM model used, including the number of parameters at each layer.



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 18, 100)           62400

 lstm (LSTM)                 (None, 18, 64)            42240

 layer_normalization (Layer  (None, 18, 64)            128
 Normalization)

 lstm_1 (LSTM)               (None, 18, 64)            33024

 layer_normalization_1 (Lay  (None, 18, 64)            128
 erNormalization)

 lstm_2 (LSTM)               (None, 18, 64)            33024

 layer_normalization_2 (Lay  (None, 18, 64)            128
 erNormalization)

 lstm_3 (LSTM)               (None, 64)                33024

 layer_normalization_3 (Lay  (None, 64)                128
 erNormalization)

 dense (Dense)               (None, 256)               16640

 layer_normalization_4 (Lay  (None, 256)               512
 erNormalization)
```

3.Training Accuracy:

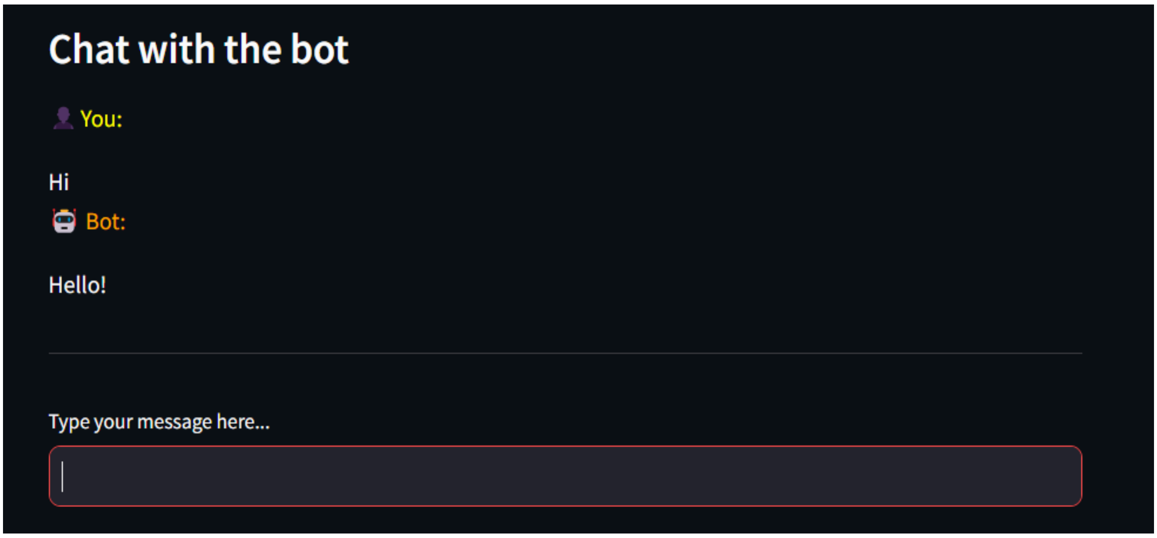
- The model training process includes early stopping based on accuracy with a patience of 3 epochs.



```
Epoch 1/100
65/65 [==============================] - 6s 17ms/step - loss: 4.9749 - accuracy: 0.0450
Epoch 2/100
65/65 [==============================] - 1s 16ms/step - loss: 4.2627 - accuracy: 0.1039
Epoch 3/100
65/65 [==============================] - 1s 15ms/step - loss: 3.8838 - accuracy: 0.1163
Epoch 4/100
65/65 [==============================] - 1s 15ms/step - loss: 3.5361 - accuracy: 0.1705
Epoch 5/100
65/65 [==============================] - 1s 15ms/step - loss: 3.0424 - accuracy: 0.2481
```

4.Response Generation:

- The function generate_answer is designed to take user input, preprocess it, predict the intent, and return a corresponding response from the dataset.



Chat with the bot

👤 You:

Hi

🤖 Bot:

Hello!

6.2 Discussion

1. **Model Performance**:
   - The LSTM model's performance is largely dependent on the quality and size of the training data. In this case, the intents dataset should have diverse and well-formed patterns and responses for optimal performance.
   - The use of multiple LSTM layers and LayerNormalization layers helps the model to capture temporal dependencies in the text data and maintain stability during training.

2. **Accuracy and Generalization**:

    - The early stopping mechanism helps prevent overfitting by monitoring the accuracy and stopping training if the model's performance does not improve for a specified number of epochs.
    - The final accuracy achieved can be an indicator of how well the model is able to generalize to unseen data. Higher accuracy signifies better generalization, while lower accuracy may indicate the need for more data or further tuning of the model.

3. **User Interaction**:

    - The chatbot is capable of engaging with users by predicting their intents based on the input text and providing appropriate responses.
    - The preprocessing steps, such as removing non-alphabet characters and converting text to lowercase, ensure that the input text is in a consistent format, which improves the accuracy of the model

CHAPTER-7

CONCLUSION & FUTURE SCOPE

5.1 Conclusion :

The Personal Motivation Chatbot project successfully addresses the critical need for accessible and personalized emotional support for students and individuals facing depression by leveraging advanced NLP techniques and LSTM-based RNN models. Through meticulous data collection and analysis, user-centered design, and the integration of powerful tools like Python and TensorFlow/Keras, the chatbot offers tailored motivational messages, real-time emotional support, and a user-friendly interface designed with Streamlit. Key features such as chat refresh, previous prompts, and daily affirmations enhance user engagement and provide timely assistance. With identified future developments, including mood tracking, CBT exercises, enhanced automation, deeper analysis, and multilingual support, the chatbot demonstrates significant potential to democratize mental health support and improve emotional resilience and overall well-being for a diverse user base.

5.2 Future Scope :

The future scope of the personal motivation chatbot project for students and individuals facing depression includes several avenues for expansion and improvement to further enhance its effectiveness and efficiency. Here are some potential areas of future development:

1. Integration of Additional Tools:

- **Mood Tracking:** Integrate mood tracking tools to help users monitor their emotional states over time and provide personalized feedback based on mood trends.
- **Cognitive Behavioral Therapy (CBT) Exercises:** Incorporate CBT exercises and techniques to offer users practical strategies for managing their thoughts and behaviors.
- **Real-Time Crisis Intervention:** Add tools for real-time crisis intervention, such as automated alerts to mental health professionals or emergency contacts during severe depressive episodes.

2. Enhanced Automation:

- **Advanced Neural Networks:** Implement more advanced neural networks, including enhanced LSTM and RNN architectures, to improve the chatbot's ability to analyze user inputs, personalize responses, and predict future emotional states.
- **Natural Language Understanding (NLU):** Utilize NLU to detect nuanced emotions and provide more empathetic and contextually appropriate responses.

3. Deeper Analysis Capabilities:

- **Sentiment Analysis:** Develop sentiment analysis tools to provide comprehensive insights into user interactions and emotional states.
- **Predictive Analytics:** Use predictive analytics to anticipate and address potential depressive episodes, offering proactive support.

4. Advanced Personalization Capabilities:

- **Machine Learning:** Continuously refine personalization algorithms using machine learning to enhance the chatbot's ability to learn from user interactions and adapt to individual needs.
- **Adaptive Learning:** Implement adaptive learning techniques to offer more tailored motivational messages and coping strategies.

5. Automation and Orchestration:

- **Routine Mental Health Support Tasks:** Automate routine tasks such as regular check-ins, mood tracking reminders, and follow-up messages to ensure consistent support.
- **Orchestration Platforms:** Implement orchestration platforms to streamline these processes and improve efficiency.

6. Real-Time Emotional Support:

- **Real-Time Chat Functionality Enhancements:** Improve the real-time chat functionality to provide more immediate and relevant support.
- **Daily Affirmations and Tips:** Expand the daily positive affirmations and mental health tips to include a broader range of supportive content.

7. Gamification and Engagement Strategies:

- **Reward Systems:** Develop reward systems to motivate users to interact with the chatbot regularly and achieve mental well-being goals.
- **Interactive Activities:** Incorporate interactive activities and games that promote mental well-being and engagement.

8. Multilingual and Multicultural Support:

- **Language Support:** Expand the chatbot's capabilities to support multiple languages to cater to a diverse user base.
- **Cultural Sensitivity:** Ensure the chatbot's content is culturally sensitive and relevant to users from different backgrounds.

9. Collaboration with Mental Health Professionals:

- **Expert-Reviewed Content:** Partner with mental health professionals to integrate expert-reviewed content, ensuring the chatbot's recommendations are clinically sound and effective.
- **Continuous Feedback:** Establish a feedback loop with mental health professionals to continuously improve the chatbot's effectiveness and reliability.

REFERENCES

1.1 Books and Articles on Mental Health and Technology:

- National Institute of Mental Health. (2021). "Depression Basics." Retrieved from NIMH.
- Kazdin, A. E., & Rabbitt, S. M. (2013). "Novel models for delivering mental health services and reducing the burdens of mental illness." *Clinical Psychological Science*, 1(2), 170-191. doi: 10.1177/2167702612463566.
- Firth, J., Torous, J., Nicholas, J., Carney, R., Pratap, A., Rosenbaum, S., & Sarris, J. (2017). "The efficacy of smartphone-based mental health interventions for depressive symptoms: a meta-analysis of randomized controlled trials." *World Psychiatry*, 16(3), 287-298. doi: 10.1002/wps.20472.

1.2 Research on Chatbots and Mental Health:

- Fitzpatrick, K. K., Darcy, A., & Vierhile, M. (2017). "Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (Woebot): a randomized controlled trial." *JMIR Mental Health*, 4(2), e19. doi: 10.2196/mental.7785.
- Inkster, B., Sarda, S., & Subramanian, V. (2018). "An empathy-driven, conversational artificial intelligence agent (Wysa) for digital mental well-being: real-world data evaluation." *JMIR mHealth and uHealth*, 6(11), e12106. doi: 10.2196/12106.

1.3 Technical Literature on NLP and LSTM Models:

- Hochreiter, S., & Schmidhuber, J. (1997). "Long short-term memory." *Neural computation*, 9(8), 1735-1780. doi: 10.1162/neco.1997.9.8.1735.
- Graves, A. (2012). "Supervised sequence labelling with recurrent neural networks." *Studies in Computational Intelligence*, 385. Springer, Berlin, Heidelberg. doi: 10.1007/978-3-642-24797-2.
- Olah, C. (2015). "Understanding LSTM Networks." Retrieved from Colah's blog.

1.4 Tools and Frameworks for Development:

- Python Software Foundation. (2021). "Python Language Reference, version 3.9." Retrieved from Python.
- TensorFlow. (2021). "An end-to-end open source machine learning platform." Retrieved from TensorFlow.
- Streamlit Inc. (2021). "Streamlit: The fastest way to build and share data apps." Retrieved from Streamlit.