Project report
on

CHATBOT FOR MENTAL HEALTH SUPPORT

Project report submitted in partial fulfillment of the requirement for the award of the Degree of
BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
Submitted By

N.KALYANI
[R180707]
V.VISWANATHA
[O180543]
A.SASI KIRAN
[R180475]

Under the Esteemed Guidance of

Mrs.P.Udayasree, Assistant Professor

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES(AP IIIT)

R.K Valley, Vempalli, Kadapa(Dist) – 516 330

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2020-2024

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES(AP IIIT)

R.K Valley, Vempalli(M), Kadapa(Dist) – 516330

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2020-2024

CERTIFICATE

This is to certify that the project report entitled “***CHATBOT FOR MENTAL HEALTH SUPPORT***” being submitted by **N.KALYANI(R180707), V.VISWANATHA(O180543), A.SASIKIRAN (R180475),**under my guidance and supervision and is submitted to **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING** in partial fulfillment of requirements for the award of Bachelor of Technology in computer science and engineering during the academic year 2022-2023 and it has been found worthy of Acceptance According to the requirements of the University.

Signature of Internal Guide **Signature of HOD**

Signature of External Examiner

ACKNOWLEDGEMENT

I wish to express our sincere thanks to various personalities who were responsible for the successful completion of the main project.

I am thankful our Director **Prof.Dr. A.V.S.S KUMARA SWAMY GUPTA ,** for providing the necessary infrastructure required for my project.

I am thankful to our Dean **Mrs. CH.RANTHNA KUMARI**, for permitting and encouraging me in doing this project.

I am grateful to **Dr.Mr. P.RAVIKUMAR(Asst.prof.ME,PHD),** Head of the Department, for his motivation and encouragement in completing the project in specified time.

I express my deep felt gratitude to **Mrs.P.Udayasree Assistant Professor,** internal guide for her valuable guidance and encouragement which enabled me to successful completion of project in time.

I express my sincere thanks to all other faculty members of CSE Department for extending their helping hands and valuable suggestion when in need.

Finally, my heartfelt thanks to my parents for giving me all I ever needed to be a successful student and individual. Because of their hard work and dedication, I have had opportunities beyond my wildest dreams.

WITH SINCE REGARDS
N.KALYANI(R180707)
V.VISWANATHA(O180543)
A.SASI KIRAN(R180475)

DECLARATION

Hereby declare that this project work entitled ***“CHATBOT FOR MENTAL HEALTH SUPPORT”*** submitted to **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING** is a genuine work carried out by me, for the fulfillment of Bachelor of Technology in the Department of Computer Science & Engineering during the academic year 2020-2024 under the supervision of my project guide **Mrs. P.UDAYASREE Assistant Professor,** Department of **Computer Science & Engineering** in **RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES(AP IIIT), R.K.Valley** and that it has not formed the basis for the award of any degree/diploma or other similar title to any candidate of the university.

WITH SINCERE REGARDS

N.KALYANI(R180707)
V.VISWANATHA(O180543)
A.SASI KIRAN(R180475)

ABSTRACT

A Chatbot is a programme that uses natural language processing (NLP) and Machine learning (ML) techniques to analyse and understand the user's query and provide with intelligent responses that are humanlike.Conversational agents that act as an intermediate between the user and the computer. The chatbot therapist is designed to provide relief to people suffering from mental illness (anxiety, depression or stress) with daily conversations and motivating them for better mental health. These therapies are mostly free of cost and available anytime. People open up more in an online chat rather than a face to face conversation. There is a need for the mechanism of using chatbots to make people understand about the options the technology is providing with and to maintain user motivation regularly.

Many articles and papers have been made that suggest using a chatbot for efficient therapy. The different conversational agents currently available in the field of pscychiatry and their role in treating ,analyzing and diagnosis of mental illness.

According to a study by the World Health Organization ,29% of the 15,000 Mental Health apps focus on the mental health.Chatbots are viable as well as an efficient method of getting mental help via your device or some may call it "reducing barriers to therapy".The most obvious things about these bots a convenience ,you can reach out to these clever bots and get help at any point in time.One of the main challenges is to design a domain specific conversational interface experience. The chatbot user interface is a bit complicated because with conversation comes expectation and to meet these expectations the design patterns need to be personalized to fit in different domains. The Chatbot is a great platform for anyone and is widely available anytime through an interface and internet connection.

CONTENTS

TITLE 1

CERTIFICATE 2

ACKNOWLEDGEMENT 3

DECLARATION 4

ABSTRACT 5

CONTENTS 6-7

CH.NO . TITTLE NAME

1. INTRODUCTION 8

1.1 INTRODUCTION 8

2. 2.1 Existing System 9

2.2 Proposed System 9

2.2.1 Introduction

2.2.2 System Architecture 9

2.2.3 Natural Language Processing 9

2.2.4 Features and Functionality 10

2.2.5 Future Enhancements 10

3. REQUIREMENT ANALYSIS

3.1 System Requirement 11

3.1.1 Hardware Specification 11

3. I .2 Software Requirement 11

3.1.3 Software Libraries & Frameworks Required 11

3.2 Software Specification

3.2.1 Python 12

3.2.2 Python Idle 12

3.2.3 An Interactive Interpreter 13

3.2.4 A File Editor 13

3.3 Libraries Used

3.3.1 NUMPY 14

3.3.2 NLTK 15

3.3.3 PYTORCH 15

3.3.4 JSON 16

3.4 Algoritms Used

3.4.1 NLP 16-17

4. METHODOLOGY

4.1 Methodology used steps 18

4.1.1 Load and preprocess the dataset 18

4. I .2 Divide the data into training dataset and testing dataset 18
4.1.3 Tokenize and pad the text data 18
4.1.4 Define the NLP model architectures 19
4.1.5 Compile the model 20
4. I .6 Train the model 20
4.1.7 Evaluate the model 20
4.1.8 Save the trained model 20
4.1.9 Load the trained model 20
4.1.10 Make predictions on new data 20
4.2 Source code 21-26
4.3 Results 27-28

5. CONCLUSION AND REFERENCES
5.1 Conclusion 29
5.2References 30-31

CHAPTER - 1

INTRODUCTION

1.1 INTRODUCTION

In recent researches, according to the World Health Organization, there are around 264 million people who suffer from depression and in the worst-case scenario, it tends to suicide. Many cases of stress and anxiety are among the people aged between 15 —29 years and the source of depression can be diverse among these age groups. Stress release is something that should be practiced by every individual. Another issue is about 76-85% of people in the lowmiddle income countries don't have proper resources or lack trained health care providers. The national health services struggle to keep up with the growing need of resources.

While developed countries have 9 mental health providers out of 100,000 people and low- income countries have as little as 0.1 for every 100,000 people.Chatbots could be a realtime saviour for such people and the only requirement is network connectivity and you have your bot the rapist ready. Therefore, we have these self-mental healthcare techniques using chatbot for stress reduction and user motivation. The therapist chatbot is the advanced method for therapy. It contains different modules that perform together to make a chatbot functional.

The inputs are provided to the chatbot based on which it identifies the level of stress according to which it gives a response to the user by providing constructive thoughts and offers self-help techniques. Using NLP (Natural language processing), the program detects how the user is feeling and then the system shows the content based on his current mood.

CHAPTER - 2

2.1 Existing System

There are several existing systems for mental health chatbots that aim to provide support and assistance to individuals. These chatbots often use natural language processing techniques to understand and respond to users' concerns. They may offer resources, coping strategies, or even provide a listening ear for those seeking emotional support. Some popular mental health chatbot systems include Woebot, Wysa, and Replika. These chatbots are designed to be accessible and convenient, providing a safe space for users to express their emotions and receive guidance. It's important to note that while these chatbots can be helpful, they are not a substitute for professional mental health care.

2.2 Proposed System

2.2.1 Introduction:

The purpose of our mental health chatbot is to provide support and assistance to individuals who may be experiencing mental health challenges. These chatbots are designed to offer a safe and confidential space where users can express their feelings, ask questions, and receive guidance. They can also offer empathy, validation, and non-judgmental support, helping users feel heard and understood. While chatbots cannot replace professional help, they can complement existing mental health services by providing accessible and immediate support.

2.2.2 System Architecture:

To develop our Mental Health Chatbot we used the technologies like NLP and frameworks like

Pytorch,nltk.

2.2.3 Natural Language Processing:

Here we used NLP techniques like:

1. Tokenization: The input is divided into smaller units, such as words or phrases, called tokens.This helps in understanding the structure of the text.

2. Part-of-speech (POS) tagging: Each token is assigned a part-of-speech tag, such as noun, verb, or adjective. This helps in understanding the grammatical structure of the input.
3. Named Entity Recognition (NER): The input is analyzed to identify and extract named entities, such as names, locations, or dates. This helps in understanding specific information mentioned by the user.
4. Sentiment Analysis: The input is analyzed to determine the emotional tone or sentiment expressed by the user. This helps in providing appropriate responses and support.
5. Intent Recognition: The input is analyzed to identify the user's intention or purpose behind the message. This helps in understanding what the user wants or needs.
6. Dialog Management: The chatbot keeps track of the conversation history to maintain context and provide relevant responses based on the user's previous messages.

2.2.4 Features and Functionality:

i)Emotional Support: Providing empathy and understanding through compassionate responses can help users feel heard and validated.

ii)Active Listening: The chatbot should be able to actively listen and understand the user's concerns and emotions without judgment.

iii)Self-Care Tips: Recommending self-care practices and activities that promote well-being, such as exercise, journaling, or engaging in hobbies, can empower users to take care of their mental health.

2.2.5 Future Enhancements:

i)Progress Tracking: Chatbots can help users track their progress over time, providing reminders for self-care activities, encouraging goal setting, and celebrating achievements.

ii) Assessment and Screening: Chatbots can use screening tools to assess mental health symptoms and provide feedback or recommendations based on the user's responses.

CHAPTER - 3

REQUIREMENT ANALYSIS

3.1 System Requirement

System Requirement Specification is a fundamental document, which forms the foundation of the software development process.. An SRS is basically an organization's understanding (in writing) of a customer or potential client's system requirements and dependencies at a particular point in time (usually) prior to any actual design or development work. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time. The SRS also functions as a blueprint for completing a project with as little cost growth as possible. The SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it.

3.1.1 Hardware Specification

- Processor: intel core i3 or above
- Processor speed: 500Mhz or above
- RAM: 4GB or above

3.1.2 Software Requirement

- Python

Python IDLE

3.1.3 Software Libraries & Frameworks Required

- Numpy
- NLP(Natural Language Processing)
- Nltk
- Pytorch

3.2 Software Specification:

3.2.1 Python

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. . It supports multiple programming paradigms, including structured (particularly, procedural),object- oriented and functional programming. Python is often described as a "batteries included" language due to its comparative standard library.

3.2.2 Python Idle

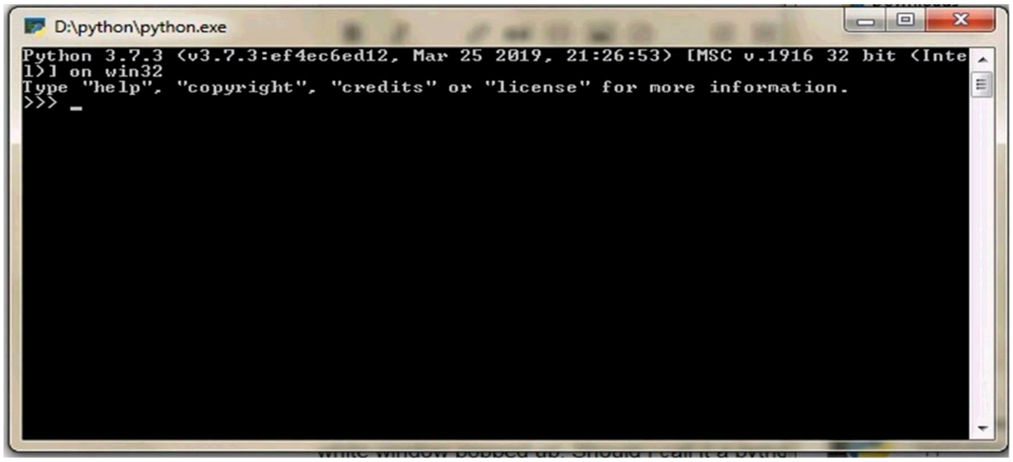
Every Python installation comes with an Integrated Development and Learning Environment, which you'll see shortened to IDLE or even IDE. These are a class of applications that help you write code more efficiently. While there are many IDES for you to choose from, Python IDLE is very bare-bones, which makes it the perfect tool for a beginning programmer.

IDLE (short for Integrated Development and Learning Environment) is an integrated development environment for Python, which has been bundled with the default implementation of the language since 1.5. 2bl. It is packaged as an optional part of the Python packaging with many Linux distributions.

12

3.2.3 An Interactive Interpreter

The best place to experiment with Python code is in the interactive interpreter, otherwise known as a shell. The shell is a basic Read-Eval-Print Loop (REPL). It reads a Python statement, evaluates the result of that statement, and then prints the result on the screen. Then, it loops back to read the next statement. Python is one of the most versatile programming languages used by developers ,data analysts,and other professionals.At the heart of this incredible language is the Python Interpreter.



```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Inte
l)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

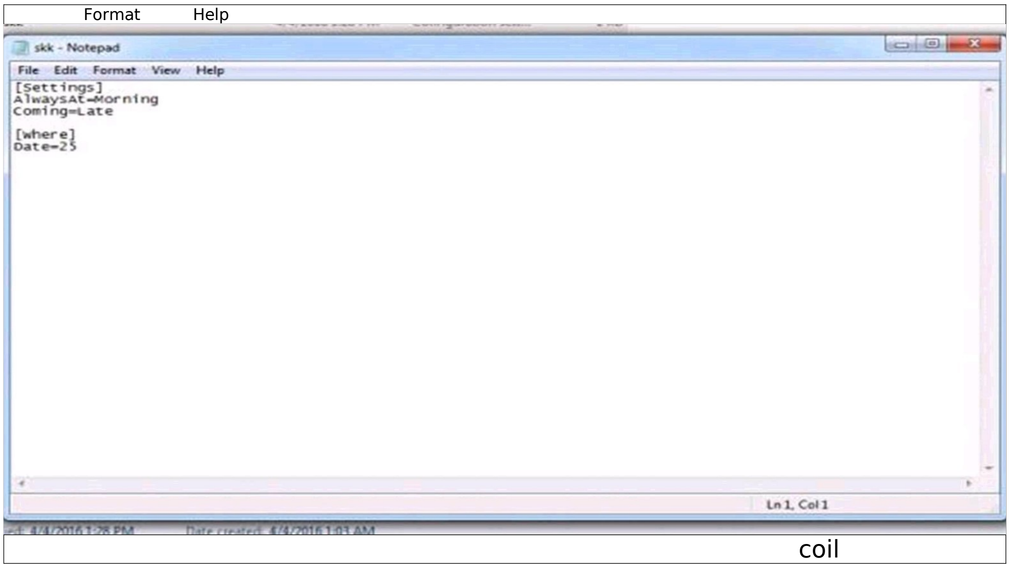
3.2.4 A File Editor

Every programmer needs to be able to edit and save text files. Python programs are files with the .py extension that contain lines of Python code. Python IDLE gives you the ability to create and edit these files with ease.

Python IDLE also provides several useful features that you'll see in professional IDES, like basic syntax highlighting, code completion, and auto-indentation. Professional IDES are more robust pieces of software and they have a steep learning curve. If you're just beginning your Python programming journey, then Python IDLE is a great alternative.

- Notep»d

Format Help



```
[Settings]
AlwaysAt=Morning
Coming=Late

[where]
Date=25
```

coil

3.3 Libraries Used

3.3.1 NUMPY

NumPy is a powerful numerical computing library in Python that provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. It is an essential tool for scientific computing and data analysis in
Python. Here are some key features and aspects ofNumPy:

i)Arrays: The fundamental data structure in NumPy is the numpy.ndarray which is a multidimensional array. These arrays can be one-dimensional, two-dimensional, or even more.

ii)Array Operations: NumPy provides a wide range of mathematical operations that can be performed on arrays, such as element-wise addition, subtraction, multiplication, and division.

iii)Universal Functions(ufuncs): NumPy includes a large collection of mathematical functions that operate element-wise on arrays, known as universal functions or ufuncs. These functions are vectorized, meaning they can operate on entire arrays without the need for explicit looping.

iv)Indexing and Slicing: NumPy supports powerful indexing and slicing operations to access elements or subarrays of an array.

v)Broadcasting: NumPy allows operations between arrays of different shapes and sizes through broadcasting. This is a powerful feature that simplifies code and improves performance.

3.3.2 NLTK

NLTK is a standard python library with functions and utilities for the ease of use and implementation. NLTK is a toolkit build for working with NLP in Python. It provides us various text processing libraries with a lot of test datasets. A variety of tasks can be performed using
NLTK such as tokenizing, parse tree visualization, etc.. .NLTK is widely used in natural language processing (NLP) and computational linguistics for tasks ranging from basic text processing to advanced linguistic analysis.NLTK is a comprehensive library for natural language processing and is widely used for research and development in the field. It provides tools for a variety of NLP tasks and serves as a valuable resource for learning and experimenting with language processing techniques in Python.

3.3.3 PYTORCH

PyTorch, a popular open-source machine learning library,provides robust tools for NLP tasks due to its flexibility and efficient tensor computations. Here are key features and aspects of
PyTorch:

i)Dynamic Computational Graph: One of the distinctive features of PyTorch is its dynamic computational graph. In PyTorch, the graph is constructed on-the-fly during the execution of the program. This dynamic approach makes it easier to debug and experiment with models. It also allows for more flexible and dynamic network architectures.

ii)Tensors: PyTorch uses tensors as the fundamental building blocks for computation. Tensors are multi-dimensional arrays that can represent scalars, vectors, matrices, or higher-dimensional data. PyTorch provides a rich set of operations on tensors, similar to NumPy.

iii)Neutral Network Module: PyTorch provides the torch.nn module for building and training neural networks. This module includes pre-defined layers, loss functions, and optimization algorithms. Users can define their custom neural network models by subclassing the nn.Module class.

3.3.4 JSON

The JSON format is syntactically similar to the code for creating JavaScript objects. Because of this, a JavaScript program can easily convert JSON data into JavaScript objects. Since the format is text only, JSON data can easily be sent between computers, and used by any programming language. JavaScript has a built in function for converting JSON strings into JavaScript objects: JSON.parse().JavaScript also has a built in function for converting an object into a JSON string:
JSON.stringify().

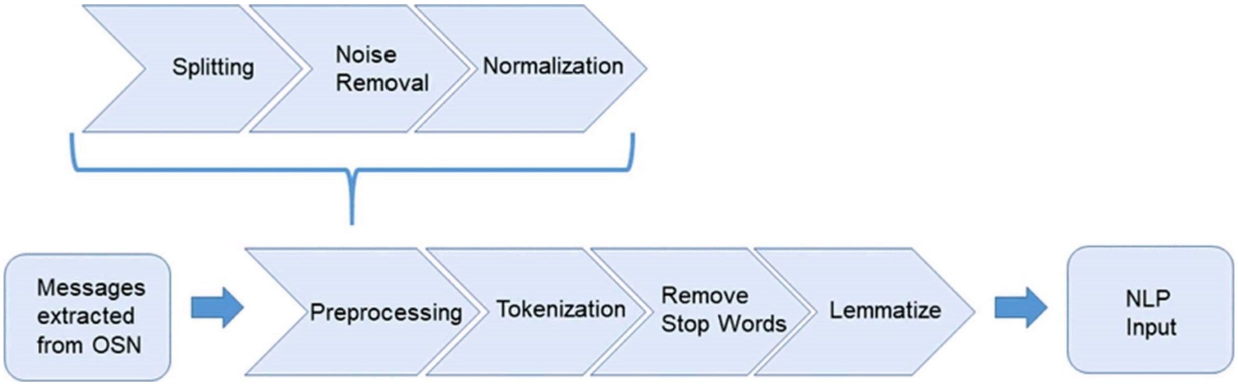
i)Universal Format: JSON is not tied to any specific programming language, making it a universal format that can be used across different technologies.

ii)Lightweight: It's a lightweight format, meaning it doesn't add a lot of unnecessary complexity to the data.

3.4 Algorithms Used

3.4.1 NLP (Natural Language Processing)

Natural Language Processing (NLP) is a branch of AI that focuses on developing computer algorithms to understand and process natural language. It allows computers to understand human written and spoken language to analyze text, extract meaning, recognize patterns, and generate new text content.



1. **Tokenization:** The input is divided into smaller units, such as words or phrases, called tokens.
This helps in understanding the structure of the text.

2.**Part-of-speech (POS) tagging:** Each token is assigned a part-of-speech tag, such as noun, verb, or adjective. This helps in understanding the grammatical structure of the input.

3.**Named Entity Recognition (NER):** The input is analyzed to identify and extract named entities, such as names, locations, or dates. This helps in understanding specific information mentioned by the user.

4.**Sentiment Analysis:** The input is analyzed to determine the emotional tone or sentiment expressed by the user. This helps in providing appropriate responses and support.

5.**Intent Recognition:** The input is analyzed to identify the user's intention or purpose behind the message. This helps in understanding what the user wants or need.

6.**Dialog Management:** The chatbot keeps track of the conversation history to maintain context and provide relevant responses based on the user's previous messages.

CHAPTER – 3

METHODOLOGY

Natural Language Processing (NLP) is a branch of Al that focuses on developing computer algorithms to understand and process natural language. It allows computers to understand human written and spoken language to analyze text, extract meaning, recognize patterns, and generate new text content. NLP models can be implemented using deep learning libraries such as
PyTorch. The code follows following series of steps:

4.1 Methodology used steps:

1.Load and preprocess the dataset
2.Divide the data into training dataset and testing dataset
3.Tokenize and pad the text data
4.Define the NLP model architecture
5.Compile the model
6.Train the model
7.Evaluate the model
8.Save the trained model
9.Load the trained model
10.Make predictions on new data

4.1.1 Load and preprocess the dataset:

The dataset is loaded from ajson file. The null values are dropped and the index is reset. The label column is converted to binary values , 1 represents matched words and 0 represents unmatched words.

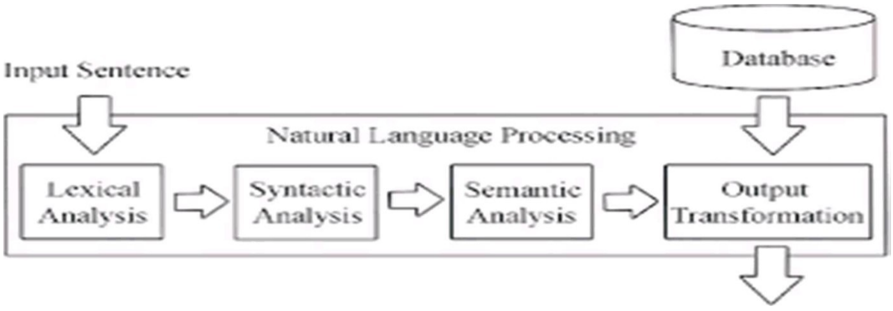
4.1.2 Divide the data into training dataset and testing dataset:

The data is allocated into 80% for training and 20% for testing.

4.1.3 Tokenize and pad the text data:

Tokenization is the process of converting text into numerical data which the neural network can process. The text is tokenized using nltk Tokenizer class and then we performed Stemming using Porter Stemmer class.

4.1.4 Define the NLP model architectures



Output Data

1.Tokenization: The input is divided into smaller units, such as words or phrases, called tokens.

This helps in understanding the structure of the text.

2.Part-of-speech (POS) tagging: Each token is assigned a part-of-speech tag, such as noun, verb, or adjective. This helps in understanding the grammatical structure of the input.

3.Named Entity Recognition (NER): The input is analyzed to identify and extract named entities, such as names, locations, or dates. This helps in understanding specific information mentioned by the user.

4.**Sentiment Analysis:** The input is analyzed to determine the emotional tone or sentiment expressed by the user. This helps in providing appropriate responses and support.

5.Intent Recognition: The input is analyzed to identify the user's intention or purpose behind the message. This helps in understanding what the user wants or needs.

6.Dialog Management: The chatbot keeps track of the conversation history to maintain context and provide relevant responses based on the user's previous messages.

4.1.5 Compile the model:

Binary cross-entropy loss function along with Adam optimizer and accuracy metric are used to compile the model.

4.1.6 Train the model:

The training set is used to train the model for a specified number of epochs.

```
Epoch [100/1000], Loss: 0.4059
Epoch [200/1000], Loss: 0.0225
Epoch [300/1000], Loss: 0.0063
Epoch [400/1000], Loss: 0.0004
Epoch [500/1000], Loss: 0.0002
Epoch [600/1000], Loss: 0.0000
Epoch [700/1000], Loss: 0.0000
Epoch [800/1000], Loss: 0.0000
Epoch [900/1000], Loss: 0.0000
Epoch [1000/1000], Loss: 0.0000
Final loss: 0.0000 training
complete. file saved to
data.path
```

4.1.7 Evaluate the model:

The model's loss and accuracy of predictions are assessed to evaluate its performance.

4.1.8 Save the trained model:

The trained model is saved in a file with a .h5 extension.

4.1.9 Load the trained model:

The saved model is loaded back into memory.

4.1.10 Make predictions on new data:

The loaded model is used to make predictions on new data, such as text. Text is tokenized and padded before being passed to the model. The output is a binary value indicating whether the result is true or false.

4.2 Source Code

File 1: nltk utils.py

```
import nltk

import numpy as np
from nltk.stem.porter import PorterStemmer
stemmer =  PorterStemmer()

def tokenize(sentence):

    return nltk.word_tokenize(sentence)

def stem(word):
    return stemmer.stem(word.lower())

def bag_of_words(tokenized_sentence, all_words):
         """
         sentence = ["hello", "how", "are", "you"]
         words=["hi","hello", "1","you","bye","thank","cool"];
         bag = [0. 1. 0. 1. 0. 0. 0]
         """

    tokenized_sentence = [stem(w) for w in tokenized_sentence]
    bag = np.zeros(len(all_words), dtype=np.float32)
    for idx, w in enumerate(all_words):
        if w in tokenized_sentence:
            bag[idx] = 1.0
    return bag
```

File 2:

```
import numpy as np
from nltk_utils import tokenize, stem, bag_of_words
import json

import torch
import torch.nn as nn
from torch.utils.data import Dataset, DataLoader

from model import NeuralNet

with open('intents.json', 'r') as f:
    intents = json.load(f)

all_words = []
tags = []
xy = []

# loop through each sentence in our intents patterns

for intent in intents['intents']:
    tag = intent['tag']
    # add to tag list
    tags.append(tag)
    for pattern in intent['patterns']:
        # tokenize each word in the sentence
        w = tokenize(pattern)
        # add to our words list
        all_words.extend(w)
        # add to xy pair
        xy.append((w, tag))

# stem and lower each word

ignore_words = ['?', '.', '!']
all_words = [stem(w) for w in all_words if w not in ignore_words]
all_words = sorted(set(all_words))
tags = sorted(set(tags))
```

Here's your code without any alterations:

python

```
import numpy as np
from nltk_utils import tokenize, stem, bag_of_words
import json

import torch
import torch.nn as nn
from torch.utils.data import Dataset, DataLoader

from model import NeuralNet

with open('intents.json', 'r') as f:
    intents = json.load(f)

all_words = []
tags = []
xy = []
```

```
# loop through each sentence in our intents patterns
for intent in intents['intents']:
    tag = intent['tag']
    # add to tag list
    tags.append(tag)
    for pattern in intent['patterns']:
        # tokenize each word in the sentence
        w = tokenize(pattern)
        # add to our words list
        all_words.extend(w)
        # add to xy pair
        xy.append((w, tag))

# stem and lower each word
ignore_words = ['?', '.', '!']
all_words = [stem(w) for w in all_words if w not in ignore_words]
all_words = sorted(set(all_words))
tags = sorted(set(tags))


# create training data
X_train = []
y_train = []
for (pattern_sentence, tag) in xy:
    # X: bag of words for each pattern_sentence
    bag = bag_of_words(pattern_sentence, all_words)
    X_train.append(bag)
    # y: PyTorch CrossEntropyLoss needs only class labels, not one-hot
    label = tags.index(tag)
    y_train.append(label)

X_train = np.array(X_train)
y_train = np.array(y_train)



class ChatDataset(Dataset):

    def __init__(self):
        self.n_samples = len(X_train)
        self.x_data = X_train
        self.y_data = y_train

    def __getitem__(self, index):
        return self.x_data[index], self.y_data[index]
    # we can call len(dataset) to return the size
    def __len__(self):
        return self.n_samples

batch_size =8

hidden_size=8
output_size=len(tags)
input_size=len(X_train[0])
learning_rate = 0.001
num_epochs = 1000

dataset= ChatDataset()

train_loader = DataLoader(dataset=dataset, batch_size=batch_size,
shuffle=True,num_workers=2)

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

model = NeuralNet(input_size, hidden_size, output_size).to(device)

# Loss and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)

for epoch in range(num_epochs):

    for(words, labels) in train_loader:
        words = words.to(device)
        labels = labels.to(device)

        #forward
        outputs =  model(words)
        loss = criterion(outputs, labels)

        #backward and optimizer step
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

    if (epoch+1) % 100 == 0:
        print (f'Epoch [{epoch+1}/{num_epochs}], Loss: {loss.item():.4f}')

print(f'final loss: {loss.item():.4f}')


data = {
    "model_state": model.state_dict(),
    "input_size": input_size,
    "hidden_size": hidden_size,
    "output_size": output_size,
    "all_words": all_words,
    "tags": tags
}

FILE ="data.pth"
torch.save(data,FILE)

print(f'training complete. file saved to {FILE}')
```

File 3:

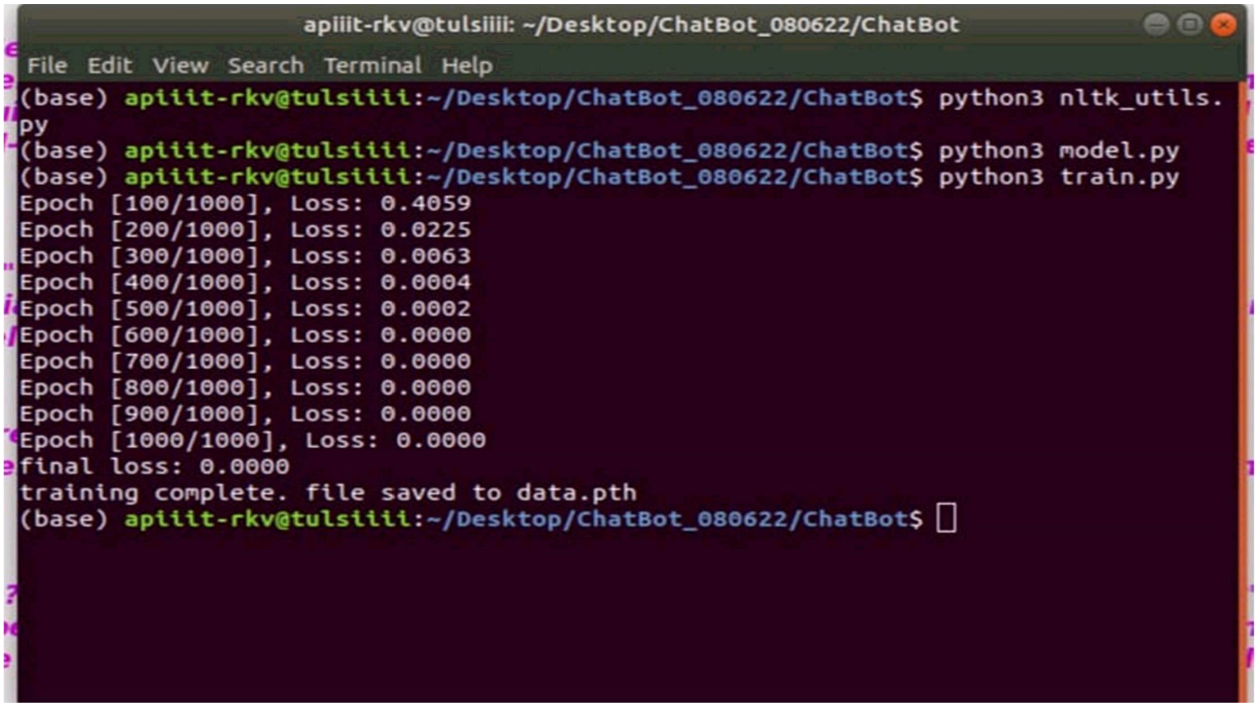
```
        import torch

        import torch.nn as nn

        class NeuralNet(nn.Module):

                def __init__(self, input_size, hidden_size, num_classes):
                super(NeuralNet, self).__init__()
                self.l1 = nn.Linear(input_size, hidden_size)
                self.l2 = nn.Linear(hidden_size, hidden_size)
                self.l3 = nn.Linear(hidden_size, num_classes)
                self.relu = nn.ReLU()

                def forward(self, x):
                out = self.l1(x)
                out = self.relu(out)
                out = self.l2(out)
                out = self.relu(out)
                out = self.l3(out)
                # no activation and no softmax at the end
                return out
```

File 4:

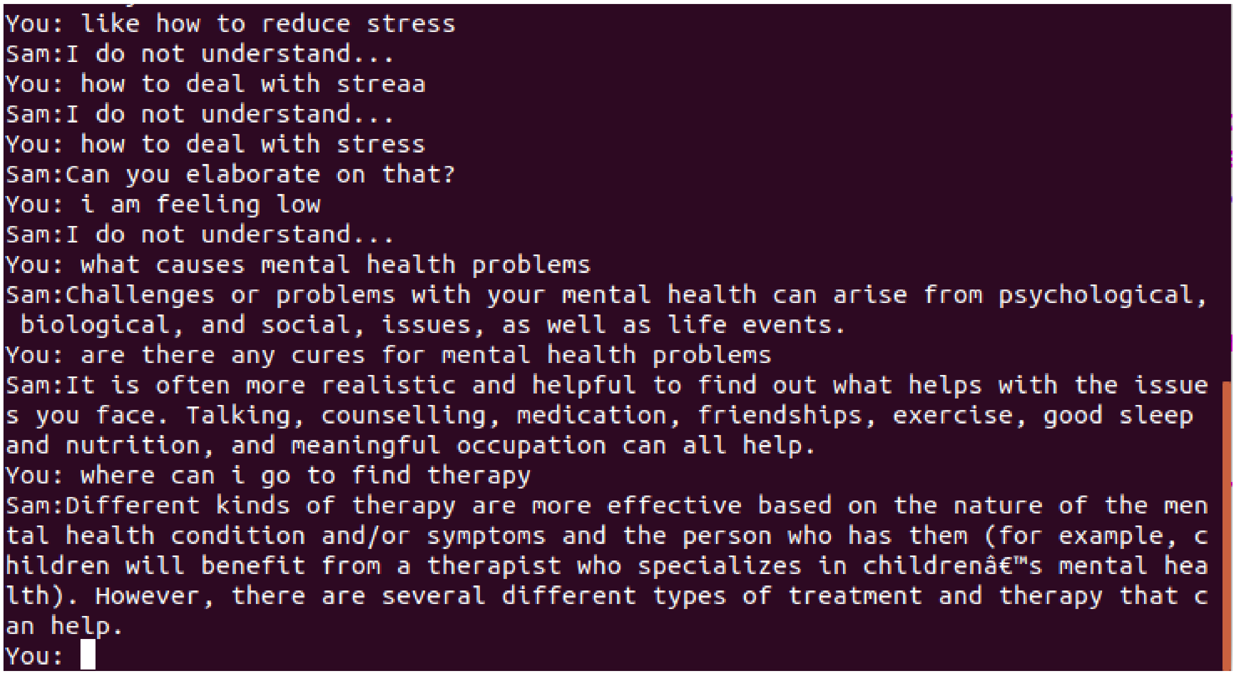
```
        import random

        import json
        import torch
        from model import NeuralNet
        from nltk_utils import bag_of_words, tokenize

        device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
```

```
with open('intents.json','r') as f:
      intents = json.load(f)

FILE = "data.pth"
data = torch.load(FILE)

input_size = data["input_size"]

hidden_size = data["hidden_size"]
output_size = data["output_size"]
all_words = data["all_words"]
tags = data["tags"]
model_state = data["model_state"]

model = NeuralNet(input_size, hidden_size, output_size).to(device)

model.load_state_dict(model_state)
model.eval()

bot_name = "Sam"

print("Let's chat! type 'quit' to exit")

while True:

    sentence = input('You: ')
    if sentence == "quit":
      break

    sentence = tokenize(sentence)
    X = bag_of_words(sentence, all_words)
    X = X.reshape(1, X.shape[0])
    X = torch.from_numpy(X)

    output = model(X)
    predicted = torch.max(output, dim=1)
    tag =tags[predicted.item()]

    probs = torch.softmax(output, dim=1)
    prob = probs[0][predicted.item()]

    if prob.item() >  0.75 :
      for intent in intents['intents']:
        if tag == intent["tag"]:
          print(f"{bot_name:{random.choice(intent['responses'])}")
    else:                   print(f"{bot_name}:I do not understand...")
```

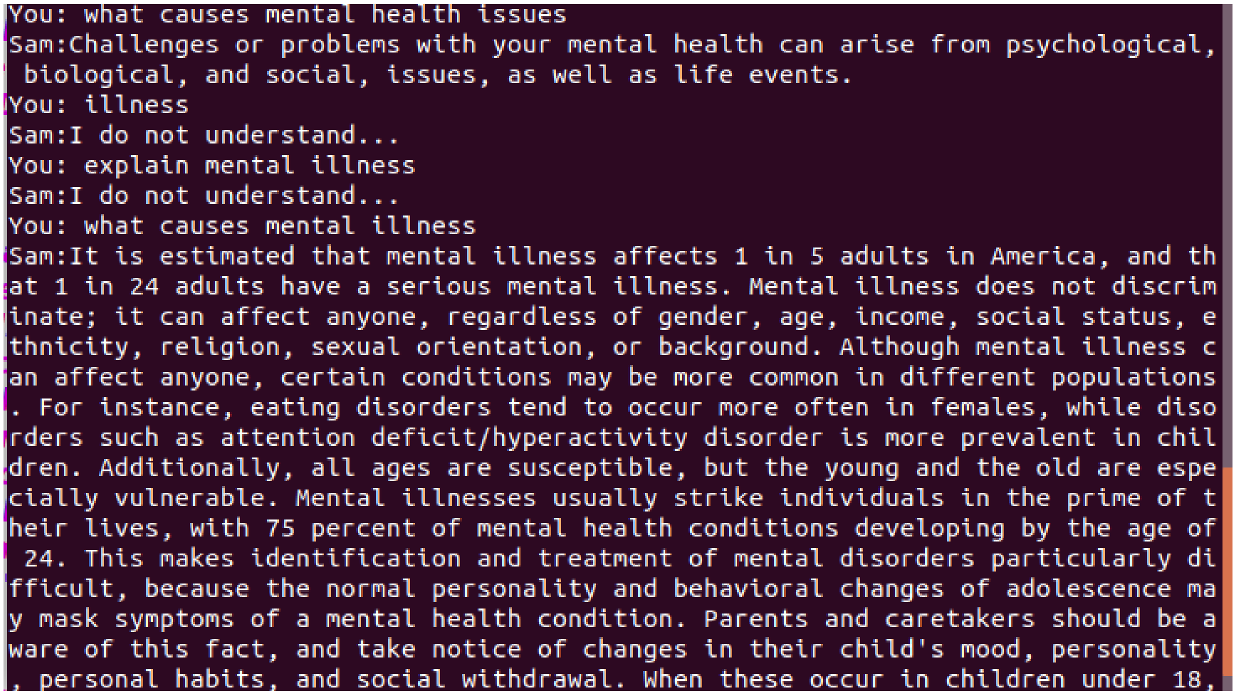
4.3 Results:


apiiit-rkv@tulsiiii: ~/Desktop/ChatBot_080622/ChatBot
File Edit View Search Terminal Help

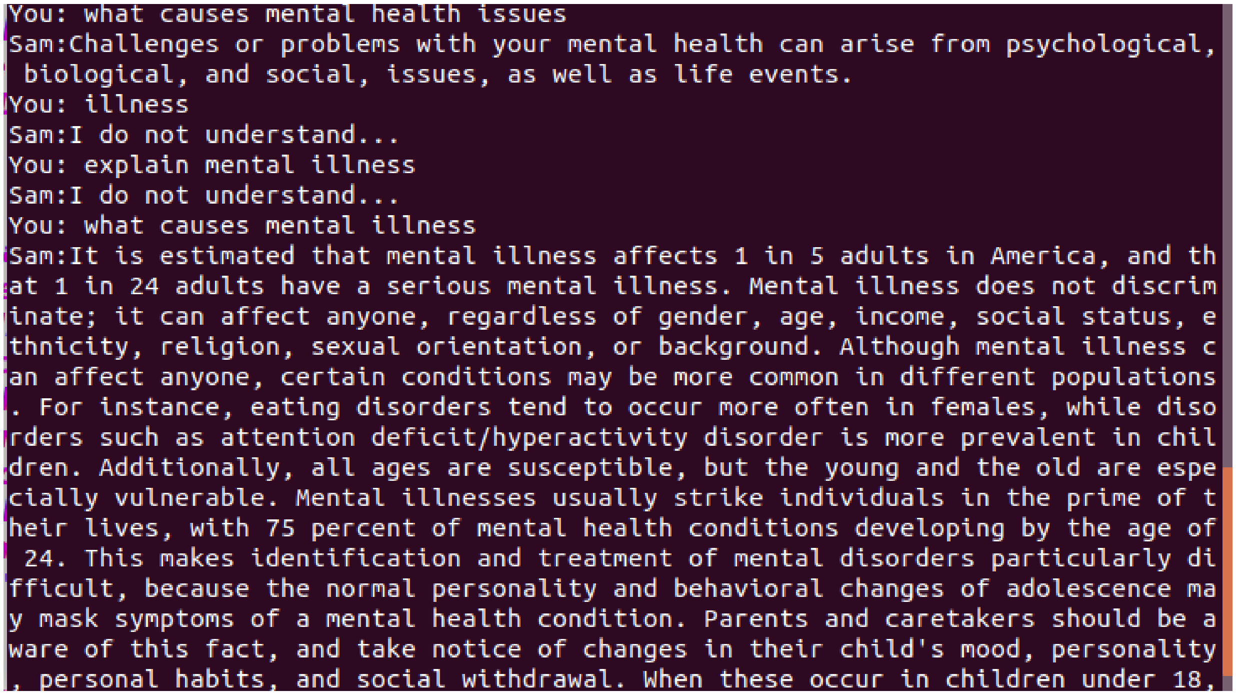
```
(base) apiiit-rkv@tulsiiii:~/Desktop/ChatBot_080622/ChatBot$ python3 nltk_utils.
py
(base) apiiit-rkv@tulsiiii:~/Desktop/ChatBot_080622/ChatBot$ python3 model.py
(base) apiiit-rkv@tulsiiii:~/Desktop/ChatBot_080622/ChatBot$ python3 train.py
Epoch [100/1000], Loss: 0.4059
Epoch [200/1000], Loss: 0.0225
Epoch [300/1000], Loss: 0.0063
Epoch [400/1000], Loss: 0.0004
Epoch [500/1000], Loss: 0.0002
Epoch [600/1000], Loss: 0.0000
Epoch [700/1000], Loss: 0.0000
Epoch [800/1000], Loss: 0.0000
Epoch [900/1000], Loss: 0.0000
Epoch [1000/1000], Loss: 0.0000
final loss: 0.0000
training complete. file saved to data.pth
(base) apiiit-rkv@tulsiiii:~/Desktop/ChatBot_080622/ChatBot$ 
```



```
You: like how to reduce stress
Sam:I do not understand...
You: how to deal with streaa
Sam:I do not understand...
You: how to deal with stress
Sam:Can you elaborate on that?
You: i am feeling low
Sam:I do not understand...
You: what causes mental health problems
Sam:Challenges or problems with your mental health can arise from psychological,
 biological, and social, issues, as well as life events.
You: are there any cures for mental health problems
Sam:It is often more realistic and helpful to find out what helps with the issue
s you face. Talking, counselling, medication, friendships, exercise, good sleep
and nutrition, and meaningful occupation can all help.
You: where can i go to find therapy
Sam:Different kinds of therapy are more effective based on the nature of the men
tal health condition and/or symptoms and the person who has them (for example, c
hildren will benefit from a therapist who specializes in childrenâ€™s mental hea
lth). However, there are several different types of treatment and therapy that c
an help.
You: 
```



```
You: what causes mental health issues
Sam:Challenges or problems with your mental health can arise from psychological,
 biological, and social, issues, as well as life events.
You: illness
Sam:I do not understand...
You: explain mental illness
Sam:I do not understand...
You: what causes mental illness
Sam:It is estimated that mental illness affects 1 in 5 adults in America, and th
at 1 in 24 adults have a serious mental illness. Mental illness does not discrim
inate; it can affect anyone, regardless of gender, age, income, social status, e
thnicity, religion, sexual orientation, or background. Although mental illness c
an affect anyone, certain conditions may be more common in different populations
. For instance, eating disorders tend to occur more often in females, while diso
rders such as attention deficit/hyperactivity disorder is more prevalent in chil
dren. Additionally, all ages are susceptible, but the young and the old are espe
cially vulnerable. Mental illnesses usually strike individuals in the prime of t
heir lives, with 75 percent of mental health conditions developing by the age of
 24. This makes identification and treatment of mental disorders particularly di
fficult, because the normal personality and behavioral changes of adolescence ma
y mask symptoms of a mental health condition. Parents and caretakers should be a
ware of this fact, and take notice of changes in their child's mood, personality
, personal habits, and social withdrawal. When these occur in children under 18,
```



```
You: what causes mental health issues
Sam:Challenges or problems with your mental health can arise from psychological,
 biological, and social, issues, as well as life events.
You: illness
Sam:I do not understand...
You: explain mental illness
Sam:I do not understand...
You: what causes mental illness
Sam:It is estimated that mental illness affects 1 in 5 adults in America, and th
at 1 in 24 adults have a serious mental illness. Mental illness does not discrim
inate; it can affect anyone, regardless of gender, age, income, social status, e
thnicity, religion, sexual orientation, or background. Although mental illness c
an affect anyone, certain conditions may be more common in different populations
. For instance, eating disorders tend to occur more often in females, while diso
rders such as attention deficit/hyperactivity disorder is more prevalent in chil
dren. Additionally, all ages are susceptible, but the young and the old are espe
cially vulnerable. Mental illnesses usually strike individuals in the prime of t
heir lives, with 75 percent of mental health conditions developing by the age of
 24. This makes identification and treatment of mental disorders particularly di
fficult, because the normal personality and behavioral changes of adolescence ma
y mask symptoms of a mental health condition. Parents and caretakers should be a
ware of this fact, and take notice of changes in their child's mood, personality
, personal habits, and social withdrawal. When these occur in children under 18,
```

CHAPTER – 5

CONCLUSION AND REFERENCES

5.1 Conclusion:

The proposed mental health chatbot can be incredibly beneficial in several ways.It can provide immediate support and guidance to individuals who may be experiencing distress or simply need someone to talk to.By offering personalized recommendations and resources,it can help individuals manage their mental well-being on a daily basis. The chatbot also creates a safe and non-judgemental space for users to discuss their mental health concerns, helping to reduce stigma and promote open conversations. It's like having a virtual companion who's always ther to listen,offer guidance,and provide helpful tools. We can chat with it anytime, day or night, to get guidance, tools, and personalized recommendations. The chatbot can also provide information and tips for our family and friends to better understand and support us. Sometimes, the chatbot can suggest lighthearted distractions or jokes to lift your spirits on difficult days, providing a momentary break. Overall, a mental health chatbot has the potential to make mental health has the potential to make mental health support more accessible, and ultimately

improve the well-being of individuals. It's an exciting and impactful project that can truly make a difference in people's live

5.2 References:

[1] https://github.com

[2] https://kaggle.com

[3] S. Bell, C. Wood, A. Sarkar, Perception of chatbots in therapy, 2019, CHI EA '19: Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems.

[4] S.H. Park, J. Choi, S. Lee, C. Oh, C. Kim, S. La, J. Lee, Designing a Chatbot for a Brief Motivational Interview on Stress Management: Qualitative Case Study, Vol 21, No 4(2019).

[5] S. Price, New chatbot therapist to help combat depression unveiled 2019.

[6] T. Kamita ,T. Ito ,A. Matsumoto, T. Munakata, T. Inoue , A Chatbot System for Mental Healthcare Based on SAT Counseling Method, 2019.

[7] Radziwill, N.M., Benton, M.C.: Evaluating quality of chatbots and intelligent conversational Agents. arXiv preprint arXiv: 1704.04579 (2017).

[8] Katherine, D., Understanding Depression. Mind (National Association for Mental Health.2001.

[9] K.Kretzschmar, H. Tyroll, G.Pavarini, Can your phone be your therapist? Young people's ethical perspectives on the use of fully automated conversational agents (chatbots) in mental health support, NeurOx young people's advisory group, 2019.

[10] A. Todorova, G. Todorova, Milen ,Digital technologies for art therapy practices used in healthcare, 2019, vol. 13 Issue 1, p43-47. sp.

[11] Egencia (2018). What is a Chatbot and How does it work? Retrieved March 9, 2019 from: https://www.youtube.com/watch?v=38sL6pADCog

[12]Wikipedia(2019).Chatbot. Retrieved March 9, 2019 from: https://en.wikipedia.org/wiki/Chatbot

[13] Chatbot Magazine (2019). A Visual History of Chatbots. Retrieved March 9, 2019 from: https://chatbotsmagazine.com/a-visual-history-of-chatbots-8bf3b31dbfb2

[14] Hattie, J. (2012). Visible learning for teachers: Maximizing impact on learning: Routledge. https://chatbotsmagazine.com/a-visual-history-of-chatbots-8bf3b31dbfb2