BUBBLE - THE AI THERAPIST

Date of submission:
5/12/2021

CSD350 NLP

Professor Ketan Bajaj

Submitted by:

Rimjhim Singh

1910110317

INTRODUCTION:

There has always been a need for emotional support and guidance, but it has become more significant since the pandemic started. People are stuck in their homes, unable to interact and express which causes immense emotional distress and hampers the emotional health of an individual. The Bubble Therapist is a step to tackle this issue. The aim of Bubble Therapist would be to talk to the users about their emotional health, hold conversation with them. Analyse their responses and give appropriate advice/support. If any red flags (severe emotional distress) are observed in the conversation with user, the AI would provide resources to seek professional help. In general cases, the AI would act as a therapist which can extend its pre-fed knowledge to guide the user away from negative thought patterns to more positive ones.

FILES NEEDED TO RUN BUBBLE:

1. main.py: This is an executable file which contains the model
2. intents1.json: This is the json file which contains our training data. This file can be swapped with any other data file that might be available.

A jupyter notebook is also attached for easy execution of the program. Contains the code in main.py and contents of intents1.json file.

MOTIVATION TO CHOOSE THIS PROJECT:

I thought about working on this project at the very beginning of the course when the App Ideation assignment was given to us. Mental health is a topic that many people in our community do not talk openly about and a lot of progress is needed in that area. This project is a little step towards relieving this situation.

WORKING OF THE MODEL:

The working of Bubble Therapist model is explained using the diagram below:

The first layer is the input data, which is the number of words that our input corpus contains. (Represented by yellow)

Then we have the two hidden neural layers. (Represented by blue)

Finally we have the output layer (Represented by green), which corresponds to the "tags" in our model. This layer is equipped with the softmax activation function. What it does is as follows: It assigns some probability to each neuron corresponding to a particular tag. The tag which has been assigned the highest probability among all, is selected and then one of the responses are selected from that tag, and shown to the user.

For example if the probability of neuron corresponding to 'greetings' tag is predicted as the highest amongst all neurons, we select it as our tag. Then one of the responses which correspond to 'greetings' is selected at random and shown to the user.

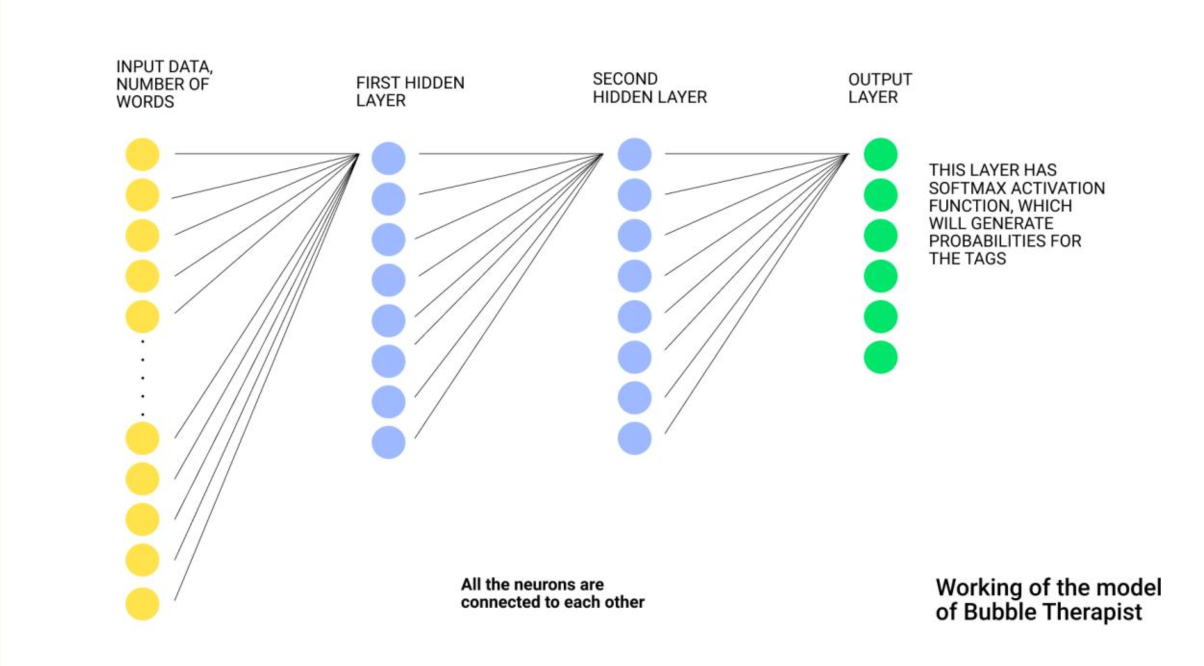
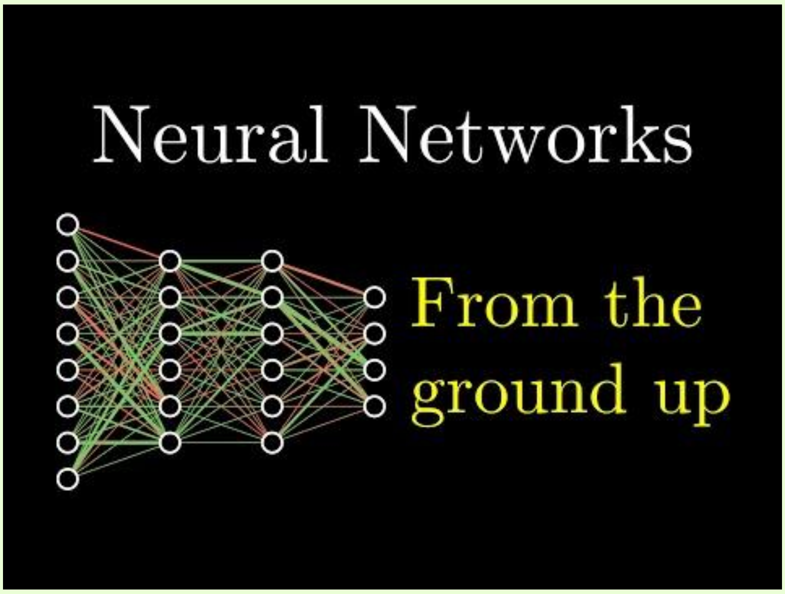

INPUT DATA, NUMBER OF WORDS
FIRST HIDDEN LAYER
SECOND HIDDEN LAYER
OUTPUT LAYER
THIS LAYER HAS SOFTMAX ACTIVATION FUNCTION, WHICH WILL GENERATE PROBABILITIES FOR THE TAGS
All the neurons are connected to each other
Working of the model of Bubble Therapist

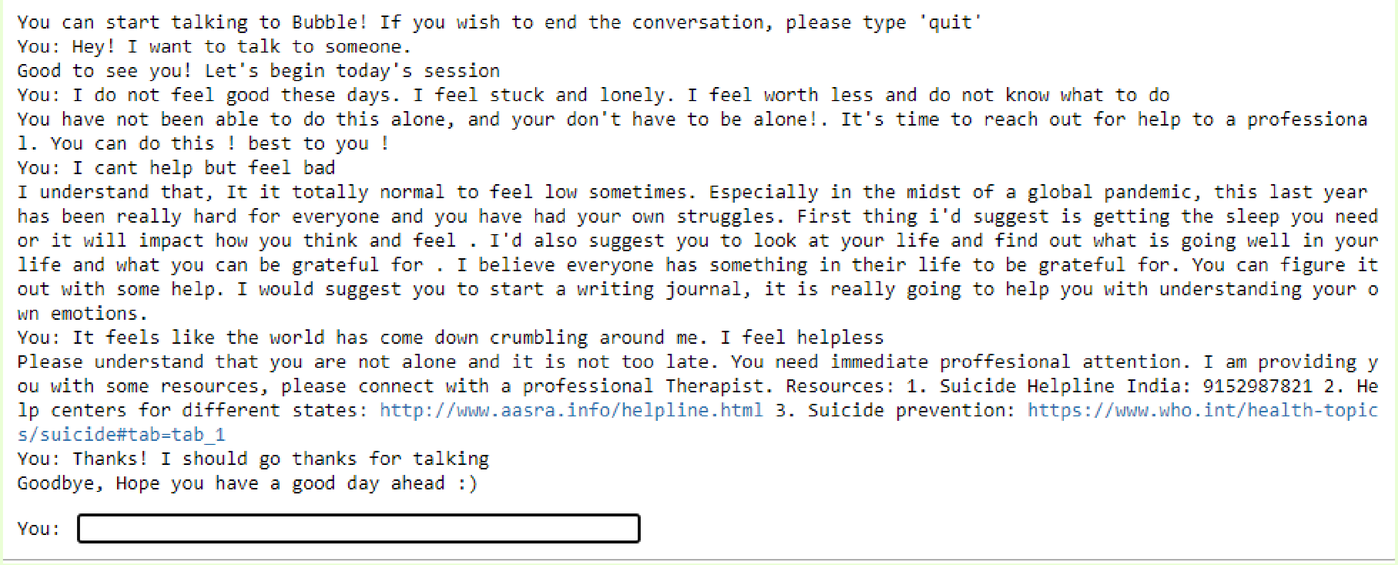
Figure 1: Model of Bubble Therapist

Note: All these layers are fully connected to each other. i.e. every neuron is connected to every other in the network.

I have linked a video on **neural networks** which really helped me understand the concept and the working of neural networks. After going through this video, I went ahead and created the graphical model of the network (Figure 1).



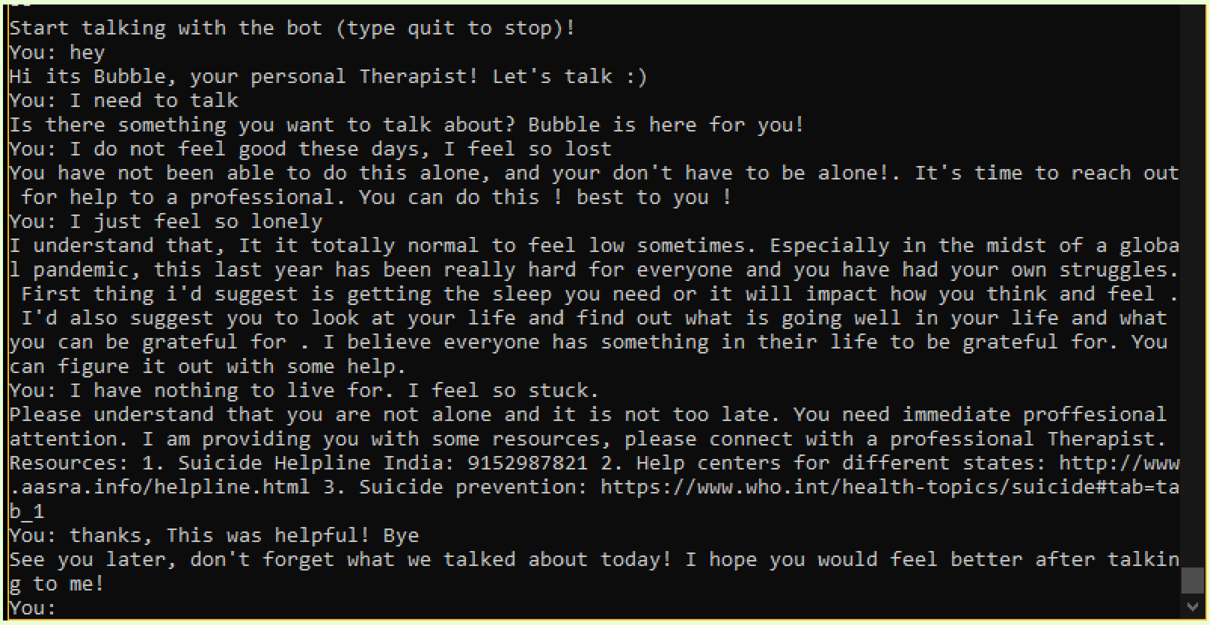
WORKING DEMO:



```
You can start talking to Bubble! If you wish to end the conversation, please type 'quit'
You: Hey! I want to talk to someone.
Good to see you! Let's begin today's session
You: I do not feel good these days. I feel stuck and lonely. I feel worth less and do not know what to do
You have not been able to do this alone, and your don't have to be alone!. It's time to reach out for help to a professiona
l. You can do this ! best to you !
You: I cant help but feel bad
I understand that, It it totally normal to feel low sometimes. Especially in the midst of a global pandemic, this last year
has been really hard for everyone and you have had your own struggles. First thing i'd suggest is getting the sleep you need
or it will impact how you think and feel . I'd also suggest you to look at your life and find out what is going well in your
life and what you can be grateful for . I believe everyone has something in their life to be grateful for. You can figure it
out with some help. I would suggest you to start a writing journal, it is really going to help you with understanding your o
wn emotions.
You: It feels like the world has come down crumbling around me. I feel helpless
Please understand that you are not alone and it is not too late. You need immediate proffesional attention. I am providing y
ou with some resources, please connect with a professional Therapist. Resources: 1. Suicide Helpline India: 9152987821 2. He
lp centers for different states: http://www.aasra.info/helpline.html 3. Suicide prevention: https://www.who.int/health-topic
s/suicide#tab=tab_1
You: Thanks! I should go thanks for talking
Goodbye, Hope you have a good day ahead :)

You:
```

Jupyter notebook implementation



```
Start talking with the bot (type quit to stop)!
You: hey
Hi its Bubble, your personal Therapist! Let's talk :)
You: I need to talk
Is there something you want to talk about? Bubble is here for you!
You: I do not feel good these days, I feel so lost
You have not been able to do this alone, and your don't have to be alone!. It's time to reach out
 for help to a professional. You can do this ! best to you !
You: I just feel so lonely
I understand that, It it totally normal to feel low sometimes. Especially in the midst of a globa
l pandemic, this last year has been really hard for everyone and you have had your own struggles.
 First thing i'd suggest is getting the sleep you need or it will impact how you think and feel .
 I'd also suggest you to look at your life and find out what is going well in your life and what
you can be grateful for . I believe everyone has something in their life to be grateful for. You
can figure it out with some help.
You: I have nothing to live for. I feel so stuck.
Please understand that you are not alone and it is not too late. You need immediate proffesional
attention. I am providing you with some resources, please connect with a professional Therapist.
Resources: 1. Suicide Helpline India: 9152987821 2. Help centers for different states: http://www
.aasra.info/helpline.html 3. Suicide prevention: https://www.who.int/health-topics/suicide#tab=ta
b_1
You: thanks, This was helpful! Bye
See you later, don't forget what we talked about today! I hope you would feel better after talkin
g to me!
You:
```

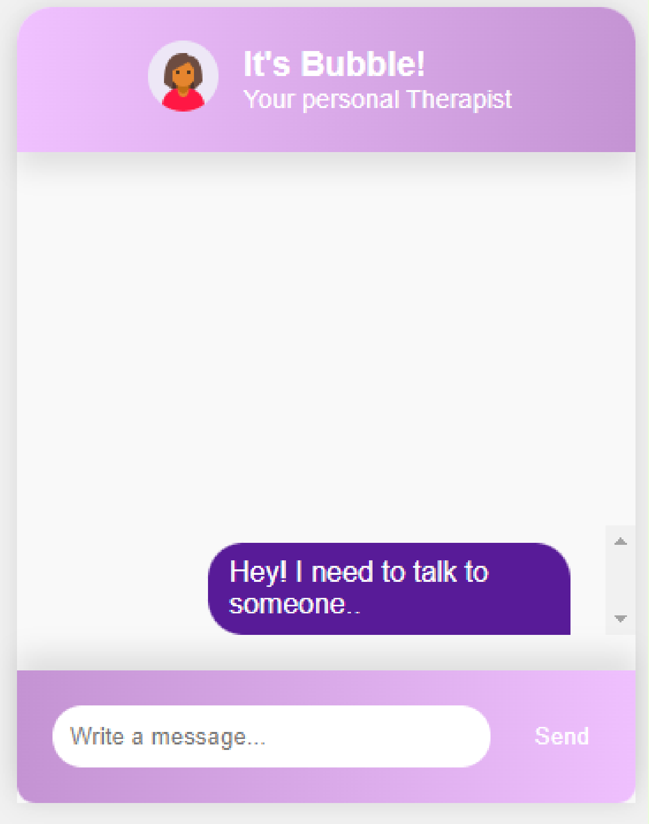
Main.py implementation on command line

ADVANTAGES

As Bubble is an AI therapist, it is highly accessible to the public - in terms of financial accessibility, freedom of expression and ease of interaction. It is user's personal Therapist that is available at the user's service 24/7 and can provide help and advice to the users whenever needed.

IMPROVEMENTS

1. The model can be improved in its accuracy.
2. More rich training data can be used to train the bot, which would increase its performance.
3. While researching about people's relationships with mental health and well-being, I came to the conclusion that a therapist or a mental health guide would work better if the user can communicate with them verbally. Actually talking about your fears and problems would be much more relieving and beneficial. On that note, in the future, a voice recognition feature could be added to Bubble. That would allow the users to record voice notes regarding their troubles and Bubble could understand them and hold conversation accordingly.
4. A frontend can be attached to the chatbot for easy interaction with user. This frontend was found to be suitable for Bubble and has to be connected to it.



It's Bubble!

Your personal Therapist

Hey! I need to talk to someone..

Code:

```
import nltk
from nltk.stem.lancaster import LancasterStemmer
stemmer = LancasterStemmer()

import numpy
import tflearn
import tensorflow
import random
import pickle

import json
with open('intents1.json') as file:
    data = json.load(file)
#print(data)

try:
    with open("data.pickle", "rb") as f:
        words, labels, training, output = pickle.load(f) #saving these variables in the file
except:
    #these blank lists are created as we want to go through the json file
    words = []
    labels = []
    docs_x = []
    docs_y = []

    for intent in data['intents']:
        for pattern in intent['patterns']:
            #tokenize the words, stemming. Root words are formed
            wrds = nltk.word_tokenize(pattern) #returns a list of tokenized words
            words.extend(wrds)
            docs_x.append(wrds)
            #gives what intent the tag is a part of
            docs_y.append(intent["tag"])

        if intent['tag'] not in labels:
            labels.append(intent['tag'])

    #removing all the duplicate elements
    words = [stemmer.stem(w.lower()) for w in words if w != "?"] #removing any question marks to not have any meaning to our model
    words = sorted(list(set(words))) #set removes the duplicate elements

    labels = sorted(labels) #sorting the labels

    #create training and testing output
    training = []
    output = []

    out_empty = [0 for _ in range(len(labels))]
    #neural network does not understand strings, but only numbers.
    #presenting the words into numbers
    for x, doc in enumerate(docs_x):
        #list to keep a check on what words are present
        #stemming the words
        bag = [] #bag of words
        wrds = [stemmer.stem(w.lower()) for w in doc]
        #going through the words and adding the information to bag
        for w in words:
            if w in wrds: #word exsits so add 1 to the list
                bag.append(1)
            else: #word does not exsit so add 0 to the list
                bag.append(0)

        output_row = out_empty[:]
        #where the tag is in our labels, and set value to 1 in output
        output_row[labels.index(docs_y[x])] = 1

        training.append(bag)
        output.append(output_row)

    #turning the lists into nparrays to be able to fed into model
    training = numpy.array(training)
    output = numpy.array(output)

with open("data.pickle", "wb") as f:
    pickle.dump((words, labels, training, output), f)

#resetting the graph data, to get rid of previous settings
tensorflow.compat.v1.reset_default_graph()
#defines the input shape for our model
net = tflearn.input_data(shape=[None, len(training[0])])
#8 neurons for the first hidden layer
net = tflearn.fully_connected(net, 8)
#8 neurons for the second hidden layer
net = tflearn.fully_connected(net, 8)
```

```
#gets probability for each neuron in the output layer,
#the neuron which has the highest probability that word is our output
net = tflearn.fully_connected(net, len(output[0]), activation="softmax") #
net = tflearn.regression(net)


#training the model
model = tflearn.DNN(net)
try:
    x
    model.load("model.tflearn")
except:
#we show the model the data 1000 times, the more times it sees the data,
the more accurate it should get
    model.fit(training, output, n_epoch=1000, batch_size=8, show_metric=True)
    model.save("model.tflearn")


def bag_of_words(s, words):
    bag = [0 for _ in range(len(words))]


    s_words = nltk.word_tokenize(s) #list of tokenized words
    s_words = [stemmer.stem(word.lower()) for word in s_words] #stemming the
words


    for x in s_words:
        for i, w in enumerate(words):
            if w == x: #if current word is equal to our word in the sentence, then
add 1 to bag list, generates the bag of words
                bag[i] = 1


    return numpy.array(bag)
```

```
def chat():
    print("Start talking with the bot (type quit to stop)!")
    while True:
        inp = input("You: ")
        if inp.lower() == "quit": #way to get out of the program
            break

        results = model.predict([bag_of_words(inp, words)]) #makes prediction, this only gives us some probability, no meaningful output
        results_index = numpy.argmax(results) #this gives the index of the greatest value in our list
        tag = labels[results_index] #maps the word to a particular tag
        #if results[results_index] > 0.6:
        for tg in data["intents"]:
            if tg['tag'] == tag:
                responses = tg['responses']

        print(random.choice(responses)) #selects a response from the tag

chat()
```