UpBeat – Your Mental Health Assistant using Rasa NLU

Aditya Rajesh Nagdekar [1], Nisha Suresh Poonia[2], Kaushik Nayan Dere [3], Dr.Amol Pande[4]

[1,2,3]UG Students, Dept. of Computer Engineering, Datta Meghe College of Engineering, Navi Mumbai, Maharastra. India
[4]Head of Department, Dept. of Computer Engineering, Datta Meghe College of Engineering, Navi Mumbai, Maharashtra, India

---***---

Abstract - *In recent years Textual conversational agent or chatbots have gained tremendous attention. In today's world, chatbots are widely used as an agent to communicate with a human be it as a booking assistant or customer service support on e-commerce and educational websites. Our aim of this project is to developed a Cognitive behavioural therapy system or a Therapy Chatbot which will be able to assess the mode of the user and also the sentiment of the interaction that it will have with user. This system or bot will also devise techniques to provide mindfulness and provide self-help to the user so that he/she can take better care of their life. In this pandemic situation there are people having emotional breakdowns, some are facing anxiety, tension, some people even get panic attacks, depression and face sleep problems. As a result, we are turning to AI solutions to help these people improve or maintain their mental health. In this world full of stressful day to day activities, no person is completely healthy because of the day-to-day stress, be it work-related stress or be it stress due to some other personal reasons. Hence, through this research, an attempt has been made to help people with depression or stress through a personal virtual psychotherapist.*

Key Words: **Psychiatric counselling, Rasa chatbot, Rasa NLU, Rasa Core, Natural Language Understanding, Postman, ngrok.**

1. INTRODUCTION

Stress is one of the major factors that are capable of influencing the mental state of a person. According to NCBI it is estimated that 14.3% of deaths worldwide, or approximately 8 million deaths each year, are related to mental disorders [1]. Hence, it can be inferred that depression, stress, anxiety has been the root cause of suicide among the masses. Hence, through this research, an attempt has been made to help people with depression through a personal virtual psychotherapist. A Chatbot is a program content which leads a discussion through sound-related or printed methods. A chatbot can cater to the normal informative needs of the user by acting as a friend or a well-wisher but a therapy chatbot can additionally cope up with the emotional needs of the user by acting as a therapist. It is also referred as online therapy. Since a depressed person does not believe in sharing his personal issues with another person for the risk of being cheated or mocked, therefore, a virtual element is best suited for the purpose of sharing.

- Some people show hesitation in sharing their personal sorrows or problems with other people because of the risk of other person judging them or them getting mocked by other person. So, in such cases, chatbots or dialogue systems can be used to get the normal informative needs of the user by acting as a friend or a well-wisher.
- Facilitation of communication: Chatbots are available 24*7 with users which resolves the issues linked to affordability and availability.
- Confidentiality is an important aspect to be kept in mind in such cases any leak of personal details can have bad repercussions so a chatbot comes to aid as personal details won't be collected.
- Increases engagement in activities: One can feel irritated and to shift the focus of mind, engaging one's mind in recreational activities reduces stigma. The more one engages himself/herself in a multitude of activities, the better he/she feels. It provides a kind of relaxation to the mind and can also be regarded as the much-needed break.
- The need for more patient-focused apps that can actually take on the role of a health care provider rather than general ones motivated us to take up the project. It is not only important but the need of the hour too but the only important thing to be kept in mind is that it must be effective.

The paper is organized as follows: Section II contains the literature survey for the project. Section III explains the Methodology. Section IV explains Data Pre-processing. Section V shows Training and Testing the model, Section VI depicts the Results. And Section VII concludes the paper.

2. LITERATURE SURVEY

A Chatbot for Psychiatric Counselling in Mental Healthcare Service Based on Emotional Dialogue Analysis and Sentence Generation published in (2017) IEEE 18th International Conference on Mobile Data Management [2] proposed a conversational service for psychiatric counselling based on of high-level natural language understanding (NLU) is suggested. Emotion recognition based on multi-modal approach is also considered. It is based on Bidirectional RNN, that is only looks text from the start as well as the end.

Digital Psychiatry – Curbing Depression using Therapy Chatbot and Depression Analysis: (ICICCT 2018) IEEE [3]-

In this paper a cognitive behavioural therapy system or a Therapy Chatbot is devised it collects the health and informative needs of a user. It detects the level of depression in a particular individual. It also suggests remedies to lower the level of depression in an individual. It looks text to determine state of user from the start of sentence.

An intelligent Chabot using deep learning with Bidirectional RNN and attention model (NCIBI 2021) [4]- In this paper Bidirectional Recurrent Neural Network (BRNN) containing attention layers is used to create the chatbot. The Bidirectional Recurrent Neural Networks (BRNN) is used so that the chatbot can give appropriate reply to user's input message which can contain 20 to 40 words and the Reddit dataset was used for training the model.

Seq-to-Seq AI Chatbot with Attention Mechanism: (arxiv.org 2020) [5]-

This paper describes an intelligent chatbot is made by implementing encoder-decoder attention mechanism architecture which uses Recurrent Neural Network with LSTM (Long-Short-Term-Memory) cells. In this an Encoder is used which creates a fixed-length vector representation of the user's input message to the chatbot and then this vector representation is used as an initial hidden state of Decoder and then Decoder generates the target response accordingly.

An Analytical Study and Review of open Source Chatbot framework, RASA (IJERT 2020) [6]

In this paper Rasa chatbot is studied, it is open source, conventional AI assistant which has two main Rasa NLU and Rasa Core. Rasa NLU acts as an ear for user inputs and Rasa Core does the processing and generates the output and gives it to the user. In this paper the implementation of Rasa chatbot has been done, Rasa uses DIET classifier which is discussed in this paper. Furthermore, how a Rasa chatbot can interact with database, APIs, Tracker Store is studied along with this different file related to Rasa such as domain.yml, config.yml are discussed in this paper.

3. METHODOLOGY

In **Fig.1** you can see various sections present in our app's architecture:

- The chat section- This section is where the user gets to chat with the chatbot.
- Meditation section- This section gives users a choice to meditate and relax themselves.
- Learn section- This section allows user to read online articles related to mental health and well-being.
- To-Do list- where the user can list down the tasks that the user wants to do throughout the day.

The meditation and article section are accessible to the user through chat section also.

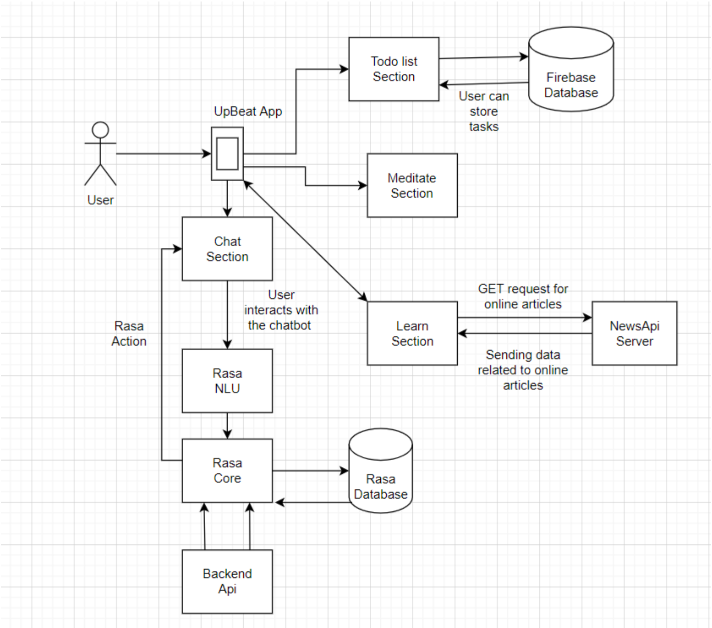


Fig 1. Application architecture

3.1 Chatbot Section:

In this section, Rasa NLU is used for intent classification and entity extraction [7]. Rasa NLU performs Natural Language Understanding, which means it takes input text and then turns it into structured data and Rasa Core performs Dialog Management, this keeps track of the conversation and decides how to proceed the conversation. Machine Learning is used by Rasa Core and NLU to learn from real example conversations.

- **Rasa NLU and Core**:

An overview of the Rasa Open-Source architecture is given below in **Fig.2**. There are two main components in Rasa, Natural Language Understanding (NLU) and Dialogue management. NLU component handles tasks like intent classification, entity extraction and response retrieval in **Fig.2** this is shown as the NLU Pipeline then there is the dialogue management component decides the next action to be performed in a conversation based on the context and in **Fig.2** this is displayed as the Dialogue Policies.

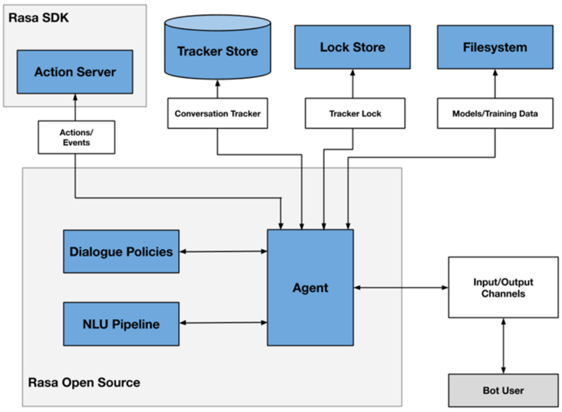


Fig 2. Rasa architecture

Rasa NLU Pipeline: The NLU Pipeline is defined in the "config.yml" in Rasa. This file contains all the steps present in the pipeline that will be used by Rasa to classify the intents and perform the appropriate action. All the components that we have used for this project are explained below.

- **Whitespace-Tokenizer:** Tokenizer uses white space as a separator it creates a token for every whitespace separated character sequence.
- **RegexFeaturizer:** It will create a list of regular expressions which are defined in the training data format during training. For each regex RegexFeaturizer creates a feature which will be set marking to check whether this expression was found in the user message or not. Then all these features will be fed into an intent classifier or entity extractor to simplify classification (assuming the classifier has learned during the training phase, that this set feature indicates a certain intent / entity). Currently only CRFEntityExtractor and the DIETClassifier components supports these Regex features for entity extraction.
- **CountVectorsFeaturizer:** Creates bag-of-words representation of user messages, intents and responses and creates features for intent classification and response selection.
- **HFTransformersNLP:** This uses pre-trained language model from HuggingFace's Transformers library and computes sequence and sentence level representations for each example in the training data by applying language model specific tokenization and featurization.
- **DIET Classifier:** By default, Rasa provides DIET (Dual Intent and Entity Transformer) Classifier, it can handle intent classification and entity extraction both. Its architecture is given below in **Fig.3**.

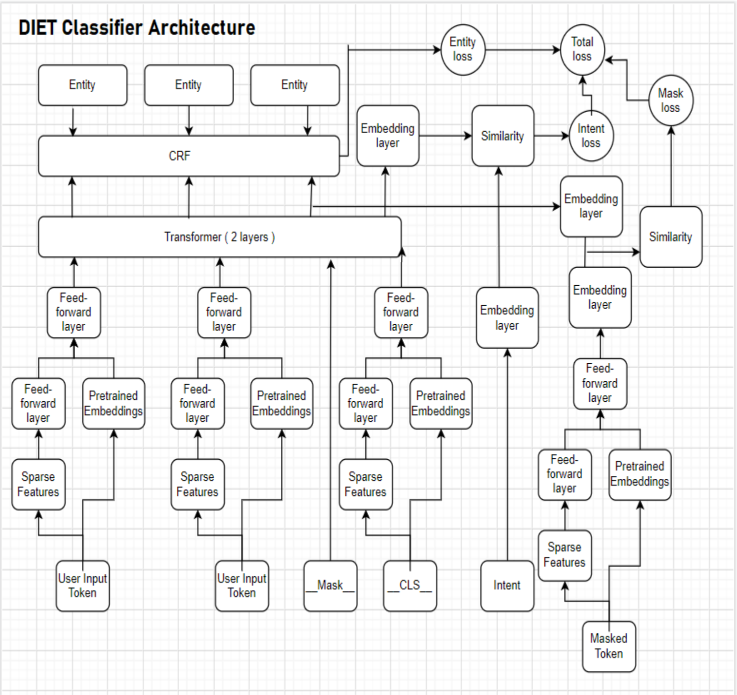


Fig 3. DIET Classifier architecture

Rasa DIET Classifier Architecture, although rasa gives full authority to the programmer to choose any classifier that he/she wants but by-default the one that rasa provides is called DIET Classifier.

- Now the user response to the chatbot is pre-processed into tokens and then it is passed on to the DIET classifier. These tokens are converted to sparse vectors using Featurizers as mentioned in the Rasa NLU Pipeline and we get a dense numeric vector of these input tokens if we are using pre-trained word embeddings like BERT, GloVe, ConVert.
- In this project we have not used any pre-trained word embeddings. The sparse vectors are passed onto a Feed Forward Layer. Rasa keeps these Feed Forward Layers sparse from the start and drops 80% of the connections. Actually, these Feed-Forward layers share same weights and they are kept sparse.
- So, from this process we get a vector which is basically a feature extracted from the input token and then it is passed to the Transformer. This process is applied on all the input tokens by DIET classifier. In this way Rasa summarizes the input tokens.
- Now for summarizing the entire input sentence by the user Rasa uses the "__cls__" token. In this, you can see that we have a Sparse Features block this block takes all the Sparse Features of the input tokens as a sum. Then there is Pre-trained Embedding's block, now this will take sentence embedding if we are using BERT.
- The output of this is a vector which is the summarization of the user input sentence which is then passed onto an Embedding layer which used to predict

the intent of the user input sentence. The output from this is matched with actual intent to find the similarity this gives us Intent Loss.

- Rasa Architecture has Transformer block at its centre this Transformer uses Attention mechanism. By-default, Rasa provides 2 layers of Transformer.
- In this project we have not used entity extraction and masking so, the total loss will only have Intent Loss. So, the Transformer is the central block which helps to find these losses which are later on optimized to improve the accuracy of the chatbot.
- Inside the Transformer all the output vectors of the Token Blocks influence and when we use the output vectors that we get from Transformer these vectors have some extra context with them. Now Let's consider the case of Intent Loss, the "extra context" comes from user input tokens and the "__cls__" token which has a lot of impact on the intent. So, the gradient that is caused by the Intent Loss gets passed to the Transformer and from there it goes to these tokens.
- The major advantage of this architecture is that you can configure it as per the needs of your project.

3.1 Meditation Section:

This section is made to help the user to build the habit of meditation every day to a soothing peaceful music which our App will provide. By meditating user can increase his/her focus and calmness. This will help user to even control his mind in difficult situations this will help the user to not panic and take firm and proper decisions.

3.2 Learn Section:

This section is developed to make user aware about one's mental health well-being, here user can read online articles about different topics related to mental health, leadership, self-improvement and positivity. This will help to create a positive attitude in the user. The online articles in this section are fetched from firing a HTTP GET request to NewsApi Api related to topics like mental-health well-being, positive attitude, leadership.

3.3 To-Do List Section:

This section is designed so that user can note down all the tasks that he/she wants to do through-out the day and keep track of them. Our main is to help user to create a proper schedule which will help user to get his work done on time. All the tasks entered in this section are stored in the Firebase Realtime Database. Firebase Realtime Database is basically, a BaaS which is known as Backend as a Service, in this project we have used firebase Realtime database and FireAuth for user authentication.

3.3.1 User Authentication and Database:

For User Authentication and Database we have used FirebaseAuth and Firebase Realtime Database respectively. Firebase Auth supports OAuth2 for Google, Facebook, Github, Twitter. It also provides the basic Email-Password method for user-authentication, in this project we have used Email-Password Authentication. Firebase Realtime Database is a NoSQL database and it it is very fast and reliable realtime database.

3.4 Connection of user with chatbot server:

In the project the chatbot is not present locally inside the mobile app but it will be running on a server. So whatever communication that will take place will be happening using HTTP GET/POST requests from mobile app to the chatbot server. So, for this purpose we have used Ngrok [9] which exposes the local server behind NATS and firewalls to the public internet over secure tunnels.

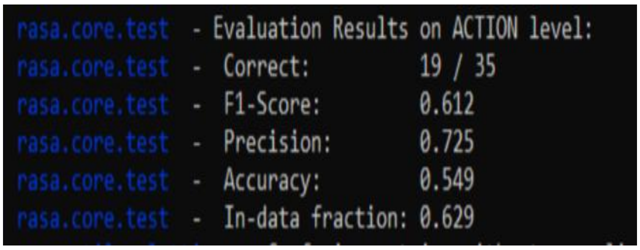
- ngrok is a cross-platform application that enables developers to expose a local development server to the Internet with minimal effort. ngrok makes locally-hosted web server appear to be hosted on a subdomain of ngrok.com this means that there is no need of public IP or domain name on the local machine.
- ngrok is able to bypass NAT Mapping and firewall restrictions by creating a long-lived TCP tunnel from a randomly generated subdomain on ngrok.com (e.g., 3gf892ks.ngrok.com) to the local machine.

4. DATA PROCESSING

We have created data which has different intent like greet, goodbye, affirm, deny, mood happy, mood unhappy among others, in a chatbot settings "**intent**" defines what the user actually wants to say to the chatbot. Different stories and actions are mentioned in separate files "**stories.yml**" and "**actions.py**" respectively. The "**stories**" defines how the chatbot should handle the conversations and "**actions**" contains all the actions that chatbot can perform and at last there is the "**policy**" which is mentioned in the "**config.yml**" file it helps to make proper predictions of actions that chatbot must perform, in this project we have used "TED policy", all of this helps to train the chatbot [10] to get desired output. The data is separated in 80-20 proportion in training and testing data set. The rules which need to be fulfilled by the chatbot are mentioned. These rules are defined to ensure that chatbot provides accurate results for certain type of queries.

5. TRAINING AND TESTING MODEL

At first while training we didn't used any pre-trained word embeddings like Bert, GloVe or ConVert. So the results of testing is given below in **Fig.4.**

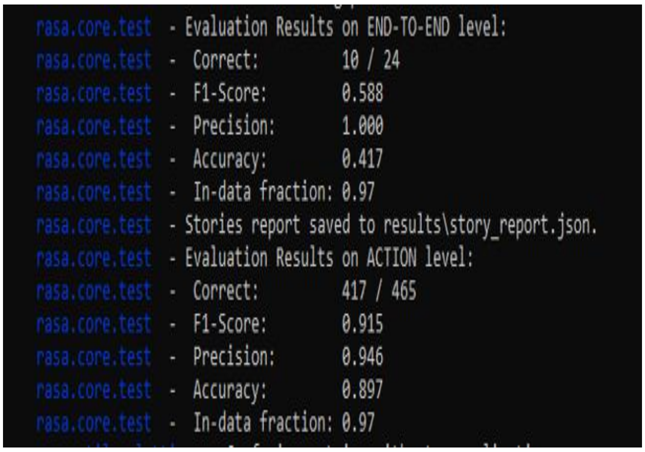


```
rasa.core.test  - Evaluation Results on ACTION level:
rasa.core.test  -  Correct:          19 / 35
rasa.core.test  -  F1-Score:         0.612
rasa.core.test  -  Precision:        0.725
rasa.core.test  -  Accuracy:         0.549
rasa.core.test  -  In-data fraction: 0.629
```

Fig 4. Testing the model without using any pre-trained word embeddings

In **Fig.4** you can see the Evaluation Results on ACTION level which means how many actions of your chatbot were properly executed out of total expected actions. At this stage our model's accuracy is 54%.

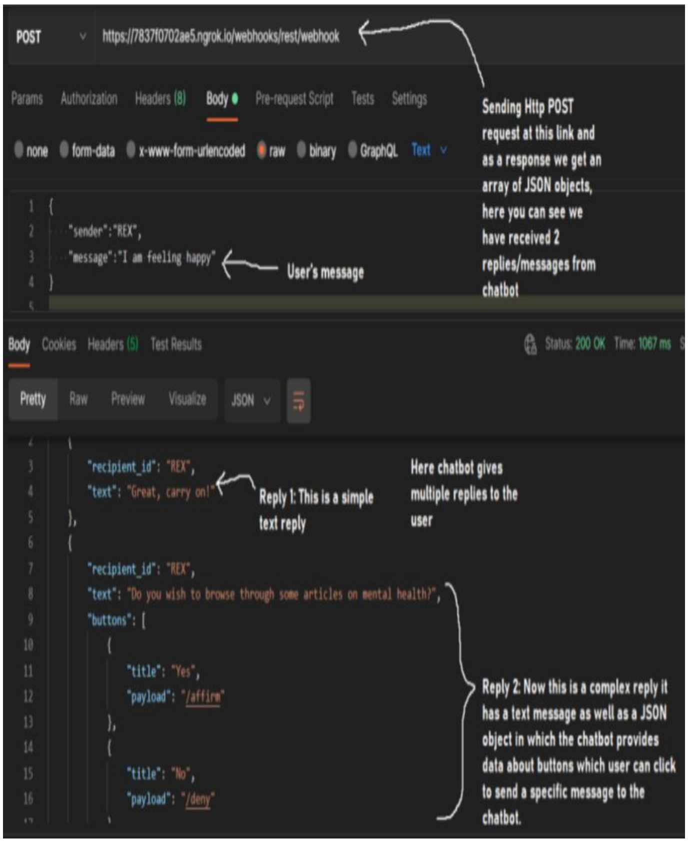
To improve the accuracy of the model we integrated pre-trained word embeddings BERT with our model. This helped us in increasing the model's accuracy, it is shown in **Fig.5.** By using model's accuracy was increased and this time we have also used Rasa's end-to-end learning, this means that now rasa can now predict what next action needs to be performed by looking at the user's input message. Rasa's end-to-end learning is a new feature which Rasa launched in their 2.2 version.



```
rasa.core.test  - Evaluation Results on END-TO-END level:
rasa.core.test  -  Correct:          10 / 24
rasa.core.test  -  F1-Score:         0.588
rasa.core.test  -  Precision:        1.000
rasa.core.test  -  Accuracy:         0.417
rasa.core.test  -  In-data fraction: 0.97
rasa.core.test  - Stories report saved to results\story_report.json.
rasa.core.test  - Evaluation Results on ACTION level:
rasa.core.test  -  Correct:          417 / 465
rasa.core.test  -  F1-Score:         0.915
rasa.core.test  -  Precision:        0.946
rasa.core.test  -  Accuracy:         0.897
rasa.core.test  -  In-data fraction: 0.97
```

Fig 5. Testing the model using pre-trained word embeddings BERT

In this we have chatbot which will be running on different operating system and user will be connected to via Android App.



https://7837f0702ae5.ngrok.io/webhooks/rest/webhook

Sending Http POST request at this link and as a response we get an array of JSON objects, here you can see we have received 2 replies/messages from chatbot

```
{
    "sender":"REX",
    "message":"I am feeling happy"
}
```

User's message

```
    {
        "recipient_id": "REX",
        "text": "Great, carry on!"
    },
    {
        "recipient_id": "REX",
        "text": "Do you wish to browse through some articles on mental health?",
        "buttons": [
            {
                "title": "Yes",
                "payload": "/affirm"
            },
            {
                "title": "No",
                "payload": "/deny"
```

Reply 1: This is a simple text reply

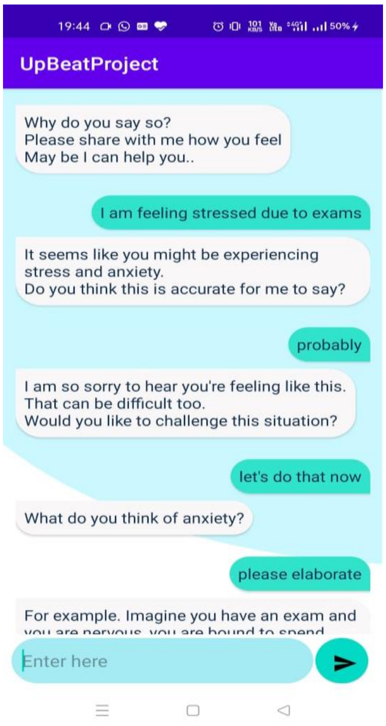
Here chatbot gives multiple replies to the user

Reply 2: Now this is a complex reply it has a text message as well as a JSON object in which the chatbot provides data about buttons which user can click to send a specific message to the chatbot.

Fig 6. Postman showing request and response from the chatbot.

So, for testing this we have used Postman tool which is used for API testing. In **Fig.6** you can see that firstly, the Http post request will be sent to URL after that you can see user's message. The chatbot responds to user by sending a reply which is basically a JSON Array. This array contains JSON object which contains the actual response. In **Fig.6** you can see the chatbot gives a text response then it also gives nested JSON object for the key "buttons", this object will have the "affirm" and "deny" buttons. On the Frontend side, this message will be shown in the form of actual buttons, an example of this is given in **Fig.7.a** and **Fig.7.b.**

6. RESULT



UpBeatProject

Why do you say so?
Please share with me how you feel
May be I can help you..

I am feeling stressed due to exams

It seems like you might be experiencing stress and anxiety.
Do you think this is accurate for me to say?

probably

I am so sorry to hear you're feeling like this.
That can be difficult too.
Would you like to challenge this situation?

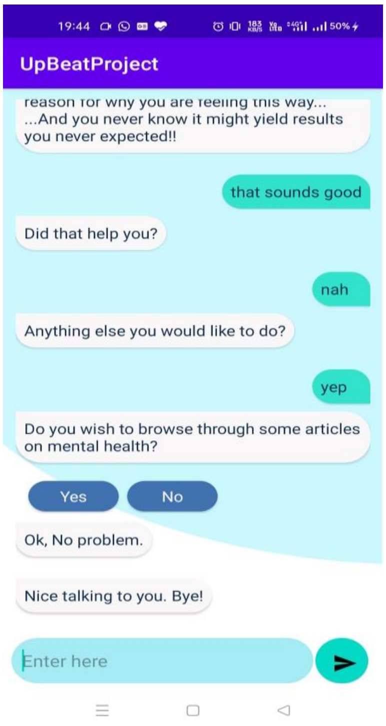
let's do that now

What do you think of anxiety?

please elaborate

For example. Imagine you have an exam and you are nervous, you are bound to spend

Fig 7.a.Chatbot section



UpBeatProject

reason for why you are feeling this way...
...And you never know it might yield results you never expected!!

that sounds good

Did that help you?

nah

Anything else you would like to do?

yep

Do you wish to browse through some articles on mental health?

Ok, No problem.

Nice talking to you. Bye!

Fig 7.b.Chatbot Section

Here you can see our project's frontend which shows the chat interface, this is what the end user will see. In **Fig 7.a** you can see a user having conversation with the chatbot regarding stress and anxiety and in **Fig 7.b** you can see the chatbot gives the user choice of whether the user wants to read some articles related to mental wellbeing and self-help in the form of buttons of "Yes" and "No".

7. CONCLUSION

In this way we have created a contextual chatbot which chats or responds to the user as per the user's mood. We have designed this chatbot in such way that it responds to user if his feeling anxious or lonely and when he is feeling happy. Through this we do not want to give the message that a chatbot will replace an expert in psychiatric counselling. This chatbot will help user to build a positive attitude towards life and breaking down the recursive cycle of negative thoughts.

8. FUTURE SCOPE

Our project successfully chats with a user and tries to understand his mood and accordingly help him lift up the mood. The future of this project depends upon the new developments in the Natural language processing field which will help in maintaining a proper flow in conversation and all the dialogues would be automated. We can increase chatbot's accuracy by increasing epochs. Epochs is basically, the number of times your machine learning model will see each training example during the training. Another thing that might help you is that you can perhaps add spaCy embeddings which will add extra features to the pipeline that might help the DIET classifier. One can use spaCy trained pipeline such as en_core_news_md or en_core_news_lg, these pipelines are for English language. We can deploy the chatbot to cloud platform by using Docker and Heroku etc.s

9. REFERENCES

1. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4461039/#:~:text=We%20estimate%20that%2014.3%25%20of,are%20attributable%20to%20mental%20disorders.

2. 2017 IEEE 18th International Conference on Mobile Data Management-A Chatbot for Psychiatric Counseling in Mental Healthcare Service Based on Emotional Dialogue Analysis and Sentence Generation.

3. International Conference on Inventive Communication and Computational Technologies (ICICCT 2018) IEEE - Digital Psychiatry – Curbing Depression using Therapy Chatbot and Depression Analysis.

4. An intelligent Chatbot using deep learning with Bidirectional RNN and attention model.

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7283081/

5. Seq-to-Seq AI Chatbot with Attention Mechanism: (arxiv.org 2020)

https://arxiv.org/ftp/arxiv/papers/2006/2006.02767.pdf

6. An Analytical Study and Review of open Source Chatbot framework, RASA (IJERT 2020)

https://www.ijert.org/an-analytical-study-and-review-of-open-source-chatbot-framework-rasa

7. https://rasa.com/docs/rasa/

8. https://firebase.google.com/products/realtimedatabase?gclid=Cj0KCQjw4ImEBhDFARIsAGOTMj9JhFpHSAT2m9XtZfwXFBnjJ2f7OGUZmWbaHbGytd6lhYk

9. https://ngrok.com/docs

10. https://rasa.com/docs/rasa/training-data-format

11. Fitzpatrick KK, Darcy A, Vierhile M. "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 Ment Health. 2017;4:e19.

1FDs3kwaAgg8EALw_wcB&gclsrc=aw.ds

https://firebase.google.com/products/storage?gclid=Cj0KCQjw4ImEBhDFARIsAGOTMjYSmjnZBHYAgXmwXgyp1i-Io8y2NW6nVQ_og8RUdNJmA8gcIbbpZgaAt12EALw_wcB&gclsrc=aw.d