CHATBOT FOR MENTAL HEALTH

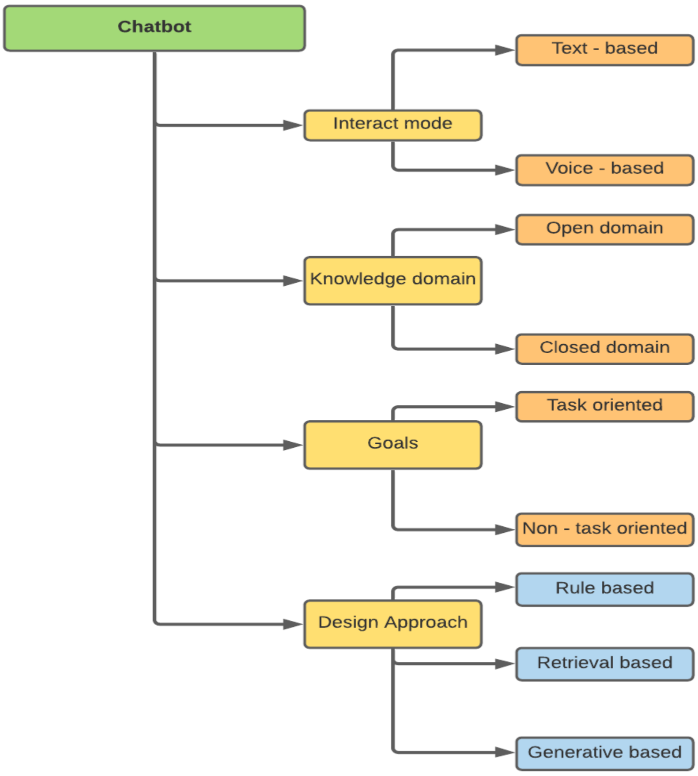
# Chatbot for mental health

This project was done for a research project. To make different kind of **chatbots from scracth in Python**, since I struggled with resources during my research.

## Motivation behind this project

In 2017, the National Mental Health Survey reported that one in seven people in India suffered from mental disorders, including depression and anxiety. The increasing awareness of mental health has made it a primary concern of development. Nearly 150 million people in India needed interventions, where the low and middle class faced more burden than the well-off people. This project is an attempt to make mental health more accessible. This conversational agent can be complemented with clinicians to make it more effective and fruitful.

## Classifications of chatbots



Chatbots can be classified on the basis of different attributes -

My research was related to the design approaches, namely, rule-based, retrieval-based, and generative-based.

1. Rule-based Chatbots: A rule-based chatbot uses a simple rule-based mapping or pattern matching to select responses from sets of predefined responses. These systems don't produce any new text; instead, they choose an answer from a predetermined list.
2. Retrieval-based Chatbots: A retrieval-based chatbot uses Machine Learning ensembles as heuristics for evaluation. Similar to rule-based chatbots, they do not generate new texts.
3. Generative-based Chatbots: Generative models do not rely on predefined responses. They come up with new replies from scratch. Machine Translation techniques are typically used in generative models, but instead of translating from

one language to another, we "translate" from input to output (response). Generative models are used for the creation because they learn from scratch.

## Overview of the bots trained

The dataset was picked up from kaggle - [Mental Health FAQ](https://www.kaggle.com/narendrageek/mental-health-faq-for-chatbot). This dataset consists of 98 FAQs about Mental Health. It consists of 3 columns - QuestionID, Questions, Answers.

Note that for training the retrieval chatbot, the CSV file was manually converted to a JSON file. Since this is not the origional dataset used for the research (read intro), I have used only first 20 rows for training the model.

The repository consists three notebooks for the three types of chatbots.

1. For rule-based, **TF-IDF** was used with **NLTK's tokenizer** for data-preprocessing. The processed data was tested against the expected outcome and **cosine similarity** was used for evaluation.
2. For retrieval-based, several Machine Learning and Deep Learning models were trained,
   - Vanilla RNN
   - LSTM
   - Bi - LSTM
   - GRU
   - CNN

Retrieval models are trained on JSON file. on For all the above models, regularization was used, and based on training and validation accuracies and loss, best model was kept for final comparisons.
It was observed that the **CNN architecture gave the best results**. The model consisted of 3 layers - convolutional neural network (CNN) + an embedding layer + fully connected layer.

3. For generative-based chatbot, NLP was used, since **NLP enables chatbot to learn and mimic the patterns and styles of human conversation**. It gives you the feeling that you are talking to a human, not a robot. It maps user input to an intent, with the aim of classifying the message for an appropriate predefined possible response.
- An encoder-decoder model was trained on the CSV file. Endoder-decoder is a seq2seq model also called the encoder-decoder model uses Long Short Term Memory- LSTM for text generation from the training corpus.
- What does the seq2seq or encoder-decoder model do in simple words? It predicts a word given in the user input and then each of the next words is predicted using the probability of likelihood of that word to occur.

### JSON vs. CSV

During this project, the biggest confusion I had was why the chatbot used a JSON file instead of CSV for the retrieval-based model. I have listed down some points that make the comparison between the two file types -
- JSON stores data in a hierarchical manner, which is better for a retrieval-based chatbot, given that the chatbot would require tags and contexts.
- A retrieval-based chatbot is trained to give the best response based on a pool of predefined answers. These predefined responses are finite in number. A tag needs to be provided for input to output mapping. To put it simply, the input given by the user(the context) is identified by the tag provided. **Based on the best tag that is predicted, the user is shown one of the predefined responses**. Hence, storing this kind of data in a JSON file is easier due to its compactness and hierarchical structure.
- A CSV file has been used for storing the data of the generative chatbot. **A generative chatbot doesn’t require tags to make predictions**. These data are easier to store in a CSV file, since we need just two columns - input text and output text. Adding or deleting data would be easier in this case as compared to a JSON file.

## Future goals

I want to research the possibilities of the generative-based chatbot further. The current encoder-decoder model cannot capture all the dependencies in the decoder layer due to the compact nature of LSTM. Attention layers can be added after LSTM layers to decode each output dynamically.