**25/08/2021**

Motivation: The goal is to build an emotion-aware conversational agent for mental health. Look for research gaps and challenges -> Address them by carrying out experimental studies -> Produce publications.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**30/08/2021**

Last sem: Literature survey

**To Do Work:**

* Read 2 research papers
* Bert tutorial
* Go through data set
* Go through Aru – bot.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**01/09/2021**

* Explore **IBM tone analyser** – It gives scores, labels/tones for the data. Also explore **MonkeyLearn** (manual labelling)
* Also look for other emotion tagging softwares
* First task: Tag the data set.

Conclusion: IBM Model is the best available.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**18/09/2021**

Data set has 4 parts – **anxiety, depression, bipolar, suicide watch.**

Part 1:

* Data set -> IBM tone analyser -> get scores for each emotion -> Check for trends -> Use clustering techniques and monkey learn.
* Trends: Frequency of each emotion in the 4 parts of data set; the mean, max, min, etc of the scores of each emotion in each part of the dataset.

Part 2:

* Understand generative, retrieval approaches of building a chatbot.
* Find mental health (Q&A) data set to generate answers (as training data for the generative model) in the chatbot.
* Check data bases which can be used to store data for the app.

**22/09/2021**

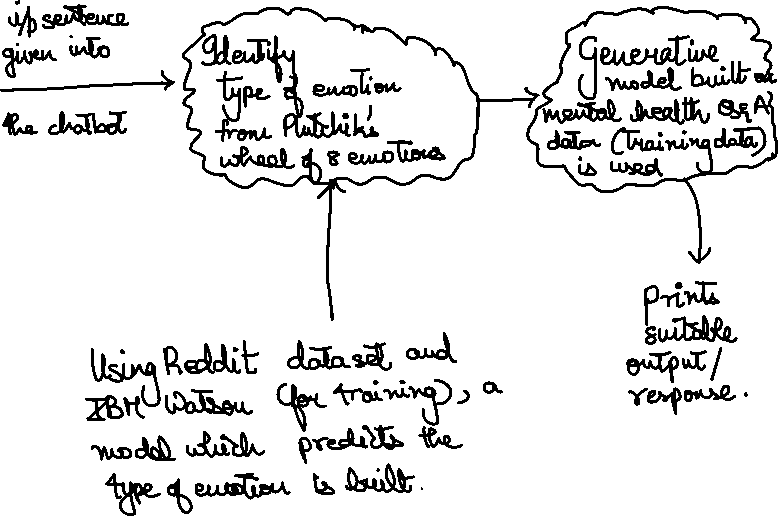
**(Understanding and exploring generative models)**

Chat bots – 2 types: 1) predefined texts and options (retrieval approach) 2) Generative approach.

We want to build a chatbot based on the generative approach as we believe that it would give better responses to mental health problems.

Generative Approach: Naïve Bayes approach is used. 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.

Work done: Read papers on how generative approaches are implemented.



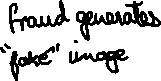
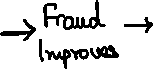
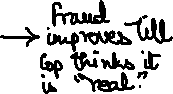
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**23/09/2021**

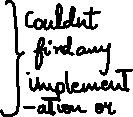
Goal: To explore generative model.

Models chosen: **GAN, BERT, GPT2, VAE, Seq2seq**. (I was asked to read and find tutorials about GAN).

* GAN = Generative Adversarial Networks.
* Mainly used in image generation softwares.



* In GAN, instead of minimizing the cross-entropy loss wrt 1-hot encoding, we try to increase the probability of the discriminator classifying the sentence as real.
* Issue with GAN text gen: -
  1. It uses RNNs (recurrent neural networks), takes previously generated tokens and hidden state as the input to generate a new hidden state. Then it uses back propagation, cross entropy loss function to predict next word. So, back propagation relies on differentiability of all layers. Unlike images, **texts are discrete.**
  2. GAN uses random sampling and so it can only score for an entire sentence.
* Solutions: -
  + Reinforcement algorithms and policy gradients
  + Gumbel SoftMax approx. (continuous approximation for SoftMax function)
  + Knowledge Distillation.



* KD-GAN: -
  + KD => knowledge distillation
  + No 1-hot encoding



* + Train a big teacher model. Student tries to mimic it.
  + Derive a continuous smooth representation of the words rather than a 1hot encoding and train the discriminator to differentiate between the representations.
  + Algorithm – given in paper.
* Seq-GAN: -
* Uses RNN, SoftMax
* Models the data generator as a stochastic policy in reinforcement learning and bypasses the differentiation problem by gradient policy update.
* Worked for Obama speeches, music generation, Chinese poems in the paper.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**29/09/2021**

* Out of the models explored, BERT was found to be the best (many resources regarding implementation are also available).

BERT – Good for context related stuff and fill ups. It is similar to seq2seq in theory.

* Next job: BERT Tutorials. Try implementing other models too.
* Plan: Use a hybrid model. Use generative model to generate sentences. But this might have syntactical/grammatical errors. So, map it to some database of sentences (retrieval) based on keywords.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**01/10/2021**

Want to implement 4 models for text generation and see how they work practically. BERT, seq2seq, Transformers XL, Gpt2 were chosen. I was asked to explore seq2seq.

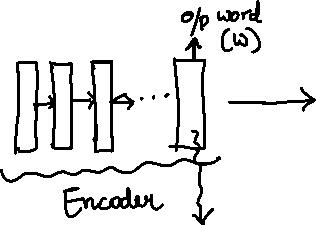
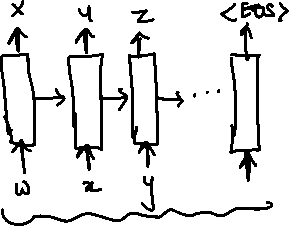
Seq2seq: -

* Initially, translations were done in a word-to-word fashion and so sentence structure was bad. Seq2seq changed this.
* Uses DL techniques and takes the neighbouring words into account.
* Input: sequence of words -> Use RNNs in the form of LSTM/GRU -> output: sequence of words.
* Encoder: Converts deep NNs and input words into hidden vectors which represent the current word and its context.

Decoder: Output of the encoder becomes the input here and prints the output.

* Applications: Translations, automated reply in mails, image captioning.
* Disadvantage: Not good for long sentences because of encoder-decoder properties.
* LSTMs:

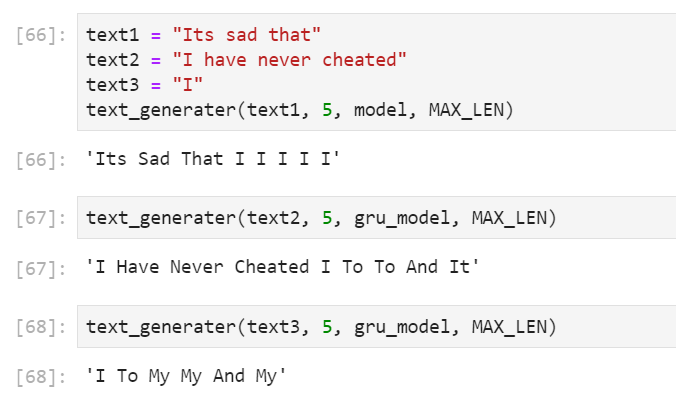
RNNs have the problem of vanishing gradient, i.e., they have only short-term memory. But in large sentences, the first word may affect the tense/singular-plural etc of the last word and so long-term memory is required. But in long-term memory, we can only store important words. When a new important word arrives, the old important word might have to be removed from memory based on its context. Hence, it is called LSTM (Long short-term memory)



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**07/10/2021**

I tried to implement the seq2seq model on the empathetic data set (Contains mental health Qs, As in alternate rows). Model was tried on both LSTM and GRU.



**Problems faced: -**

1. Seq2seq doesn’t work great for longer sentences.
2. This code was trained on only 900 sentences. Actual training data has 77000 sentences (Approx.) but since a nested-for loop is used, the computer crashes if I try to use it.

Solution: Try Google Collab pro

**Gpt2 Model** (Tried by Pritish) **gave better results** on the data set. His model was trained on the complete data set.

Next Steps: -

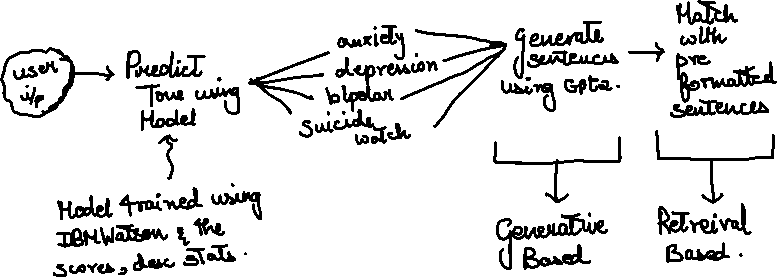
* Train seq2seq with full data set
* Try using other tutorials for seq2seq
* Research about the performance metrics to be used for comparing the text generation outputs.

Comparison should be objective (Tests like accuracy, F Measure, AUC) as well as subjective (Does the generated sentence make sense).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**27/10/2021**

Post Midsem Work: -



Work for tomorrow – Implement Gpt2 model for ‘suicide watch’ using given tutorial.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**08/11/2021**

Goal: Find Trigger Words to flag

(Get a list of trigger words, get their word embedding and train a model)

Constraints:

* + 1. We need to get the list of words from a reliable source
    2. What happens on flagging? (Send alert to family member/therapist, etc).

But response should be in accordance to ethical guidelines.