DATA641

Natural Language Processing

Final Project Write up

Topic: Mental Health Chatbot

Tausif Khan, Varun Jain, Allen Mathews

Table of contents

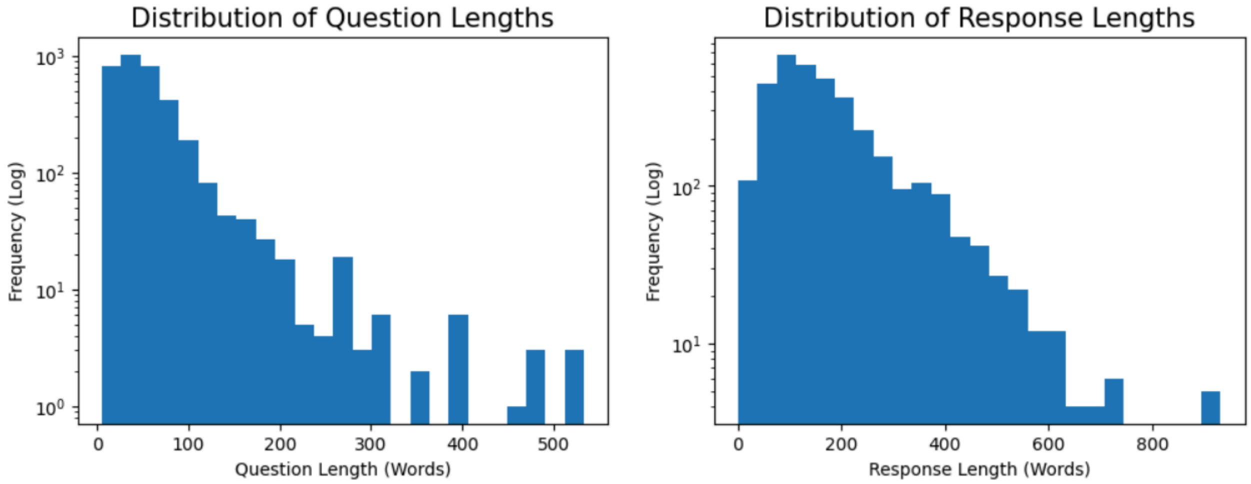
Introduction

The growing need for mental health support has become a pressing global concern, impacting people across all demographics. The rise in mental health disorders are evident, yet the available resources to address these issues are sadly insufficient. A significant challenge is the need for accessible and affordable mental health resources and professionals. This problem is severe in regions where the resources are limited. Furthermore, the high cost of mental health care makes it inaccessible to people without the financial means, which leads to uneven distribution of care and resources globally. Lack of diagnosis and treatment of mental health disorders are widespread, mainly among low income groups. They encounter obstacles, such as stigma, lack of awareness, and cultural biases within the healthcare system. Consequently, they are less likely to receive timely and appropriate care, leading to worsening conditions and significant impacts on their overall well-being. Traditional methods of delivering mental health care are inadequate for meeting the diverse and growing needs of the population. There is an urgent need for innovative strategies to improve both accessibility and affordability in mental health services, ensuring that quality care is available to everyone, regardless socioeconomic status or geographical location.

The progression of AI/ML offers promising solutions to these challenges. The goal of our project is to deploy a mental health chatbot, which provides affordable and accessible support. These chatbots can deliver immediate assistance to those in need, and help ease the burden on mental health professionals. By leveraging these technologies, we can develop a comprehensive system that addresses these challenges, enhance the accuracy and responsiveness of mental health interventions, and ultimately improve the mental wellbeing of everyone involved.

Data Source

We are dealing with data in the mental health domain. The dataset that we are using contains high-quality therapist responses to mental health questions sourced from online counseling and therapy platforms instead of using data scraped from public forum sites such as Reddit. The dataset contains 3.5k entries where the questions cover a wide range of topics and the responses are provided by qualified individuals who range in licensing from PhD level psychologists, social workers, and licensed mental health counselors. The data was scraped from a reliable website counselchat.com. There are 31 topics on the website forum ranging from issues such as “depression” to “family issues”. When looking into the dataset, we noticed that most questions are pretty short while the therapists provide longer responses. The average length of a question is around 50 words while the average length of a response is 150 words.



Data Preprocessing

Since we started off with an already cleaned dataset, we did not have to deal with duplicates or errors in the data such as typos. However, we handle missing values by dropping them in case there are any present in the dataset. We divided our data into training, validation, and test sets to evaluate the performance of machine learning models. We preprocess the data by stripping any punctuation from the text. Then, we tokenize the text and remove any non-alphabetic characters. After that, we remove stopwords using the stopwords library in NLTK. Once the stopwords are removed, we lemmatize each word to get a tokenized list of words. This cleaned and standardized text data is then ready for vectorization and training of future models.

Baseline System

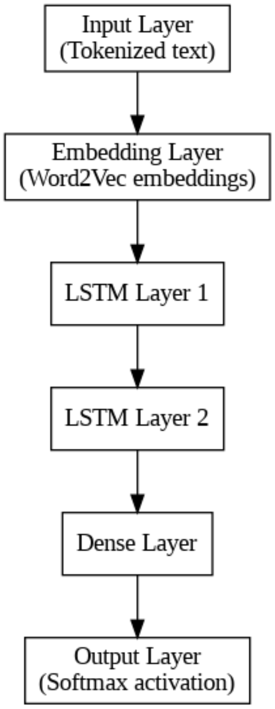
To get started with our chatbot we needed to first be able to understand specific intentions behind each user message, so to do this we started off by training a multi-class classifier to understand the topic or intention of a user’s message. First we create a validation set by manually labeling a randomly selected set of question and response pairs. We need to do this since the dataset we are using does not contain any predefined labels or annotations. We tried a couple of models at first such as K-NN, SVM, Random Forest, and K-means clustering. The classifier that gave us the best results was a SVM model on TF-IDF features for the classifier due to its advantage of being able to handle the sparsity and dimensionality of text data well. Since n-gram feature extraction did not give us too much to work with, we went with TF-IDF ultimately to help us get more interpretable models. Unfortunately, performance on the validation set is not great. One reason for that could be the way we created our labels, since we are not experts in the mental health domain we determined labels for the validation set based on our intuition. Looking at the support for each class in the validation set we see that many topics only show up one or two times, so it’s no surprise that we have basically no consistent and reliable ability to predict the correct label for a given user prompt. With more data points, these results should improve.

Model

LSTM (Long Short-Term Memory)

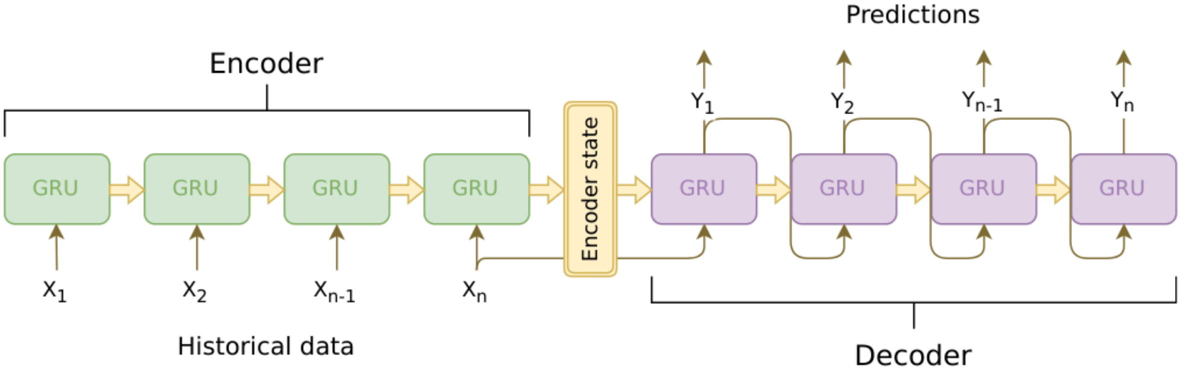
This RNN (Recurrent Neural Network) technique is mainly used to understand sequential data such as time series data, but it can also be used to analyze text and speech. It is used to capture long term dependencies in sequential data, store them as a memory cell and transmit it to the subsequent LSTM layers. There are three gates associated with this cell, the input gate (monitors new data into the cell), the forget gate (removes unnecessary data), and the output gate (manages data flow to LSTM output). In this way, the LSTMs can forget the irrelevant information, and retain the essential information for a longer time.

Since our goal is to generate an appropriate response for an input text, we train the LSTM model with the preprocessed data obtained from the previous steps. Before we train the model, we generate Word2Vec embedding for the input data. Word2Vec is used to capture meaning and context of a token using the surrounding tokens by obtaining vector representations of the input text. These input vectors go through an 80/20 train-test split. Our model consists of multiple LSTM layers for increased complexity. The embedded vectors are fed to the LSTM layer. Each word in the sequence is processed by the LSTM one at a time, by producing a hidden state for each word. The intent of the input can be predicted using these hidden states, which captures the meaning of the text. The output of these LSTM layers are fed into a dense layer which consists of a softmax activation function to ensure interpretability of the output. We train the model with 10 epochs with the objective of capturing complex patterns within the data. Once the model was trained, we ran a test input to observe the performance of the model.



Seq2Seq

To create our chatbot model, we employed a Seq2Seq (Sequence-to-Sequence) approach, which is primarily used to convert sequences of one type into another. We implemented a basic Seq2Seq model using PyTorch, training a neural network on our dataset of input-output sequence pairs. For preprocessing, we used the GPT-2 tokenizer to convert sentences into tokens, ensuring standardized input for the neural network. After tokenization, we collated the data points into batches and fed them into the neural network. During training, the model processed the inputs and expected outputs, using its forward pass to understand the context and decode appropriate responses. We trained the model over 5 epochs, during which the loss function decreased with each iteration, indicating the model's learning and improvement. This approach resulted in a more coherent and contextually aware chatbot capable of generating human-like responses.



Evaluation

We evaluate the LSTM model with some sample inputs, and analyze each output to test the performance. The evaluation is done on unseen data to ensure that the model can generalize new inputs well. The input feed text undergoes the same preprocessing, such as tokenization and lemmatization. However, the performance of the model output was less than satisfactory due to certain limitations faced during the development process. The model was tested for the following inputs:

Sample input 1: 'I'm going through some things, I cannot sleep at night. How do i change my feeling'

Output: always always influence belief inspiration change

Analysis: The response output, although imperfect in fluency, does generate some keywords such as 'belief', 'inspiration', and 'influence', which can result in intentions such as 'positive influence', 'always keeping belief', 'inspiration to change', etc, which is a positive response in the context of overcoming negative feelings as described in the text input.

Sample input 2: ‘Need help’

Output: common common

Analysis: The response output is imperfect in both fluency and adequacy, as a consequence of the performance of the model.

The performance problems seen in the LSTM model's responses underscore the difficulties in creating an effective mental health chatbot. Important factors include the quality and relevance of the training data, the limitations of computational resources, and the ethical considerations when dealing with sensitive data. Tackling these issues is crucial to enhance the model's accuracy, empathy, and overall effectiveness in delivering mental health support.

Limitations

Though the project has potential, there are several limitations to consider. Using a publicly sourced dataset encounters issues such as noise, inconsistencies, or biases that affect the quality and reliability of the models predictions. The database might not represent certain demographics, which can lead to imperfect results. RNN models, such as LSTM, demand high computational power and time. Lack of computing resources leads to longer training time, which consequently leads to less effective models. Additionally, working on mental health dataset brings up important ethical issues. It is critical to ensure privacy and anonymity of the data, lack of which could have serious consequences for the individuals whose data is being used. Limited training time and the high complexity of the data make it hard for the model to capture all necessary patterns effectively. Additionally, the model tends to be biased towards certain labels or features in the dataset. This bias often comes from imbalanced data, where some responses or categories are more represented than others. Consequently, the model may focus too much on specific parts of the data and ignore other important aspects, leading to skewed predictions and reducing the model's overall fairness and reliability.

To summarize, these limitations highlight the difficulties in creating a dependable and effective mental health chatbot. To overcome these obstacles, it is crucial to have access to high-quality, meticulously curated datasets, and to leverage the expertise of mental health professionals. Additionally, substantial computational resources are needed to train and optimize the models effectively. Adhering to stringent ethical standards is also essential to ensure the responsible application of AI in handling sensitive mental health data and providing accurate and unbiased support.

Conclusion

Throughout this project, we have faced various challenges, such as lack of computing recourse and domain knowledge. But despite these challenges, we have procured valuable insights and progress that shows the potential in providing accessible mental health support, upon which more development and improvements can be made in the future.

To run through various checkpoints in the development of the chatbot, at first, we procured the database, preprocessed the data to remove irrelevant stop words and punctuations, tokenized the data and lemmatized the tokens which are suitable for training. For a few instances, we assigned intents which highlight various types of mental health problems, ran a K-Means classification to classify the remaining data points to the previously assigned intents. We generate Word2Vec embeddings for these tokens, which provide an efficient context for the model, enabling it to generate more relevant responses. These embedded vectors are fed into the LSTM model, which consists of multiple layers, not including a dense layer, and trained for 10 epochs for added complexity. This architecture allows the model to generate responses that reflect a deep understanding of the input text. We tested the model with input texts and analyzed its outputs for its performance. Although there is room for improvement, the model's capability to produce contextually appropriate responses in test instances shows the robustness of the LSTM architecture. These tests revealed the model's strengths and pinpoints areas needing further refinement.

References

- https://huggingface.co/datasets/Amod/mental_health_counseling_conversations
- https://towardsdatascience.com/word-and-character-based-lstms-12eb65f779c2
- https://medium.com/@neri.vvo/how-to-use-lstm-in-nlp-tasks-text-classification-example-3222db759337
- https://jeddy92.github.io/ts_seq2seq_intro/