# CHAPTER 1

**INTRODUCTION**

## Problem

Mental health is incredibly important because it affects our overall well-being, relationships, and ability to function in daily life. Clear thinking, wise decision-making, and effective coping with life's challenges and stressors are all benefits of good mental health. Conversely, if mental health is poor, it can result in a variety of issues such as challenges in personal relationships, reduced productivity, physical health complications, and even self-harm or suicidal thoughts. Mental health issues can also impact our ability to function effectively in work, school, and other aspects of daily life. Recognizing the importance of mental health can help us take steps to maintain good mental health, seek help when we need it, and reduce the stigma surrounding mental health issues. By prioritizing our mental health and taking care of our emotional well-being, we can lead happier, healthier lives. Suicide is a highly significant public health concern globally, and India is no exception. Mental health issues are a significant risk factor for suicide, and the burden of mental illness in India is high.

According to the National Crime Records Bureau (NCRB) of India, there were 1,39,123 reported suicides in the country in 2019, with a suicide rate of 10.4 per 100,000 population. This is an increase from the 2018 suicide rate of 10.2 per 100,000 population. While suicide rates are high across all age groups in India, young adults between the ages of 18 and 45 are particularly vulnerable. In 2019, 60% of reported suicides in India were by individuals in this age group. The suicide rate for young adults in India is also higher compared to other age groups.

Mental health issues are a significant risk factor for suicide in India. Depression, anxiety, and bipolar disorder are among the most common mental health conditions associated with suicide in the country. Other factors that may contribute to suicide risk include social isolation, financial stress, and relationship problems. Despite the high suicide rates in India, there is still significant stigma surrounding mental health issues. Many individuals may not seek help for mental health concerns due to fear of discrimination or lack of access to mental health services.

Additionally, mental health resources in India are limited, particularly in rural areas. Efforts are being made to address the issue of suicide in India, including increasing access to mental health services and reducing stigma surrounding mental illness. The National Mental Health Survey of India, conducted in 2015-2016, highlighted the need for improved mental health services and increased awareness of mental health issues in the country. Overall, suicide rates due to mental health issues in India remain a major public health concern. Increased awareness, education, and access to mental health services are critical to addressing this issue and reducing the burden of suicide in the country.

## Background

Social media content can be an effective tool to detect suicidal thoughts because it provides a large amount of information about an individual's thoughts, emotions, and behavior. People often express their feelings and thoughts on social media platforms such as Reddit, Twitter, Facebook, Instagram, and others, and this information can provide insights into their mental health status. For example, individuals who are experiencing depression or anxiety may post messages or updates on social media that reflect their negative thoughts and emotions. They may also use specific keywords or phrases that are associated with suicidal ideation, such as "hopeless," "worthless," or "I can't go on anymore." Researchers have found that analyzing social media content can help to identify individuals who are at risk of suicide. Machine learning algorithms can be trained to analyze patterns in social media content and identify language that is associated with suicidal ideation.

Moreover, social media platforms can also play a role in identifying and preventing suicide by providing resources and support to users who are experiencing mental health issues. Some social media platforms have implemented features that allow users to report concerning content, and these reports can be directed to mental health professionals who can offer support and resources to individuals who may be at risk of suicide. It is important to note that social media content should not be used as the sole method for identifying suicidal thoughts or mental health concerns. It is also important to consider other factors, such as the individual's personal and medical history, and to encourage individuals to seek professional help when needed. However, social media can be a valuable tool in identifying individuals who may be at risk of suicide and providing support and resources to those who need it. Chatbots can be a useful tool to help suicidal people because they offer a confidential and accessible way for individuals to get support and assistance when they need it.

Here are some reasons why chatbots can be a good idea to help suicidal people, chatbots are available 24/7, which means that individuals can get support and assistance at any time of the day or night. This is particularly important for individuals who may be experiencing suicidal thoughts outside of regular business hours or who may not have access to traditional mental health services. Anonymity: Chatbots provide a level of anonymity that can make it easier for individuals to share their feelings and thoughts without fear of judgment or stigma. This can be particularly important for individuals who may be hesitant to seek help due to social or cultural barriers. Chatbots can be customized to provide personalized support and assistance to individuals based on their specific needs and concerns. This can help to ensure that individuals receive the right level of support and assistance that they need to manage their mental health concerns. Chatbots can be programmed to provide crisis intervention and support to individuals who are at immediate risk of self-harm or suicide.

This can help to ensure that individuals receive the support they need in a timely manner, which can be critical in preventing suicide. Chatbots provide consistent support and assistance to individuals, regardless of the time or day. This can help to ensure that individuals receive the same level of support and assistance every time they interact with the chatbot, which can be important for building trust and rapport.

Overall, chatbots can be a useful tool to help suicidal people because they provide a confidential, accessible, and personalized way for individuals to get support and assistance when they need it. While chatbots should not replace traditional mental health services, they can be a valuable addition to the mental health toolkit, particularly for individuals who may face barriers to accessing traditional services.

## Objective

The objective of this project is to develop a model that can predict suicidal tendencies in social media posts. Our ultimate goal is to demonstrate the practicality of our suicide detection model by incorporating it into an operational chatbot that can engage in meaningful conversations with young people and provide them with appropriate professional resources when signs of distress are detected.

# Chapter 2

**LITERATURE** **SURVEY**

**2.1. Social Media and rise of Chatbots**

Social networking sites and online networks have become increasingly popular as platforms for users to share their thoughts and feelings. Unfortunately, one of the biggest concerns today is the rise in suicides, with people often expressing their suicidal thoughts through social media platforms such as Reddit. Social media is recognized as an essential medium for self-expression, allowing users to be closely monitored and their preferences and interests recorded.

In 2022, a recent study found that approximately 300 million individuals worldwide experienced depression. There are various types of statements that may indicate suicidal thoughts. By utilizing sophisticated technologies such as Natural Language Processing (NLP) and Machine Learning (ML) Models, we can examine patterns of suicidal thoughts and behavior, and create prevention systems to avert such devastating outcomes. While most studies focus on using a single machine learning system to identify despair, researchers are currently exploring various algorithms to determine which can most accurately address this issue. A chatbot is a program that indulges in multiple interactions simultaneously. It works like the human brain using artificial intelligence, machine learning and neural capabilities. Chatbots speak as easily as humans. Chatbots caught his attention in 2016. 2016 is said to be the first year of chatbots.

More than many startups and companies have started using chatbots to improve customer service. Research shows that chatbots are now being used in railway bookings, bus bookings, stay bookings, logistics, and businesses like Amazon and Flipkart analyze common queries based on patterns. The main advantage of using his AI/ML in chatbots is that it is easy to use and learn.

The salient characteristics of chatbots are their user-friendly interface, conversational system, and capacity to operate in multiple languages, with the ability to create structured, computer-readable representations.

Many start-ups have developed numerous chatbots, which organizations have employed to enhance their customer service and provide prompt, compassionate responses. Research shows that chatbots are currently being used across a wide range of industries, including e-commerce, insurance, banking, healthcare, finance, legal services, telecommunications, logistics, retail, automotive, leisure, travel, sports, entertainment, media, and other sectors. Various organizations are now using chatbots to respond quickly and productively, and some even post occasional customer inquiries.

The utilization of therapy chatbots has the potential to revolutionize the mental health sector and provide numerous advantages to both patients and healthcare practitioners. They can improve the efficiency of mental health services by automating administrative tasks, allowing practitioners to focus on delivering personalized patient care. This can help combat burnout among healthcare professionals and improve the quality of treatments offered to mental health patients. Moreover, therapy chatbots can offer accessible and affordable healthcare support to thousands of patients who may not have access to mental health services due to several reasons. They can also provide patients with doctor-approved information and resources to self-diagnose their symptoms, thereby enabling them to manage their symptoms and seek appropriate help. Additionally, therapy chatbots can monitor patient-reported outcomes and provide insights that can help healthcare professionals better diagnose and treat mental health patients.

They can also be used to set up online appointments, thereby eliminating the need for patients to travel to hospitals and clinics to seek help.

However, it is essential to note that further research is needed before therapy chatbots can be widely adopted. The chatbots must be designed to protect the personal health information of patients, and the technology must be continually monitored and updated to ensure that it remains effective and accurate.

## 2.2. Research Gaps and Limitations

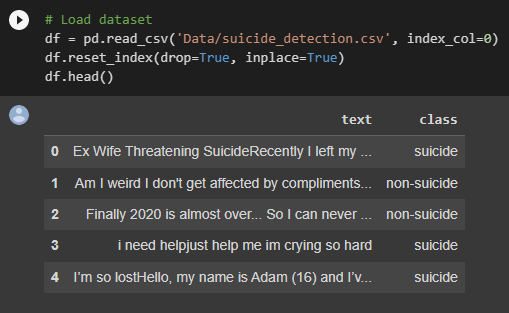
I agree that chatbots can be a valuable tool in the mental health, particularly in providing support and resources to individuals who may not have access to traditional mental health services. Nevertheless, as we have pointed out, chatbots have certain restrictions in their capacity to offer empathetic replies and effectively recognize signs of suicidal tendencies. Therefore, it is important to continue researching and developing chatbots that can provide more nuanced and effective answers. Additionally, ensuring the privacy and security of personal health information is crucial for the widespread adoption of mental health chatbots. Overall, chatbots have the potential to be a valuable complement to traditional mental health services, but careful consideration and development are necessary to ensure their effectiveness and ethical use.

# CHAPTER 3

# DATA COLLECTION & WRANGLING

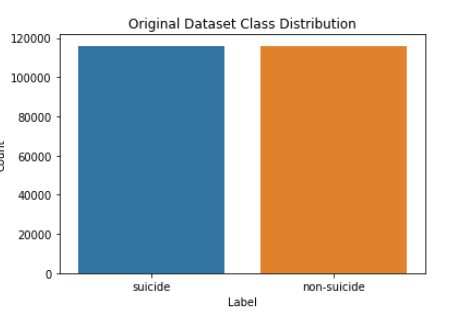
## Data Collection

The dataset utilized for detecting suicidal ideation in this project has been acquired from Kaggle. It consists of textual posts from the social media platform, Reddit.



**Fig.** **3.1** A Fraction of Dataset on Display

The dataset has been labelled using a binary classification approach, where posts from subreddits ‘SuicideWatch’ are labelled "suicidal" and posts from ‘teenagers’ are labelled "non-suicidal". It's important to note that this type of labelling approach may have some limitations, as there may be some posts that do not fit into either of these categories. Nonetheless, it is a good starting point for building a model to identify suicidal ideation in social media posts.

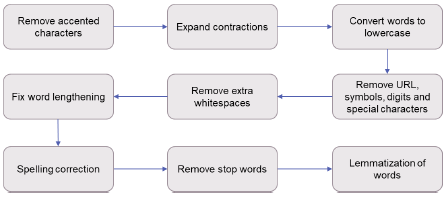


**Fig. 3.2** Original Dataset Class Distribution

The classes are equally distributed, as seen in the dataset where there are 116,037 rows, representing 50% of the dataset, within each class.

## Text Pre-processing

Prior to model building, the text data necessitates pre-processing to transform it into suitable formats. Social media data, specifically, is typically less structured and requires more tailored preprocessing and cleansing techniques. Thus, our data was cleaned with the following steps in the sequence seen in Figure below.

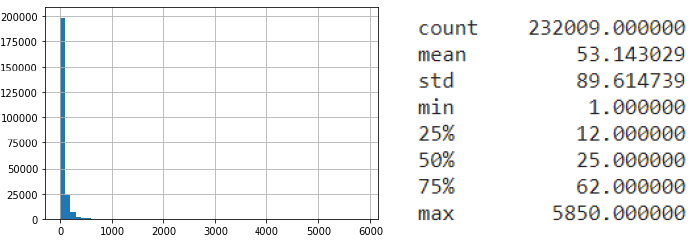


**Fig** **3.3** Text Pre-Processing in Detail

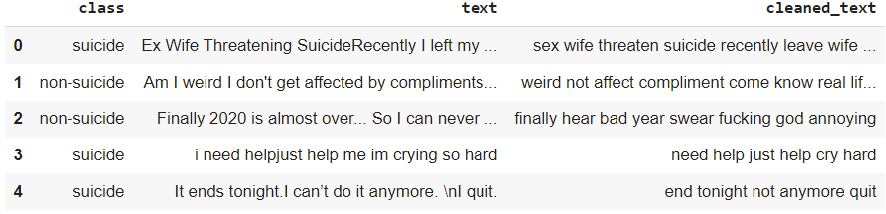
Text Pre-processing is carried out to remove accented characters, expand contractions, convert to lowercase, remove website URL(s), symbols and digits, special characters, and extra whitespaces, work on word lengthening, spelling correction, remove stop-words, lemmatization.

## Data Cleaning

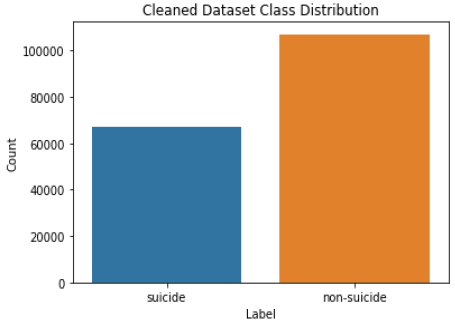
We remove empty rows and the irrelevant words post pre-processing. Mainly there is action on removing the outliers from the dataset with abnormally high word count to reduce training time and increase efficiency on smaller datasets.



**Fig** **3.4** Activity Diagram of Speech Emotion Recognition



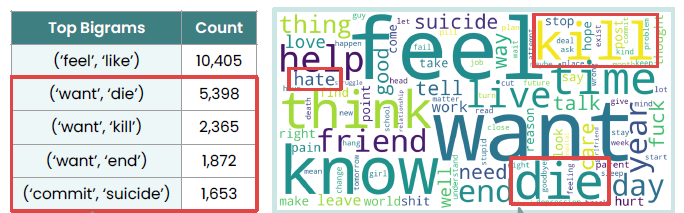
**Table 3.1** Sample rows of cleaned dataset



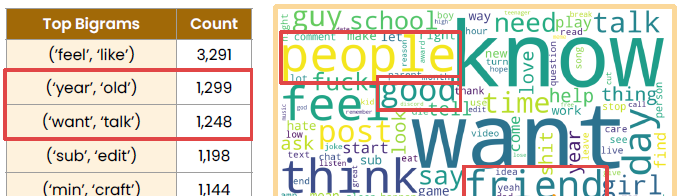
**Fig** **3.5** Cleaned dataset class distribution

**3.4. Data Exploration**

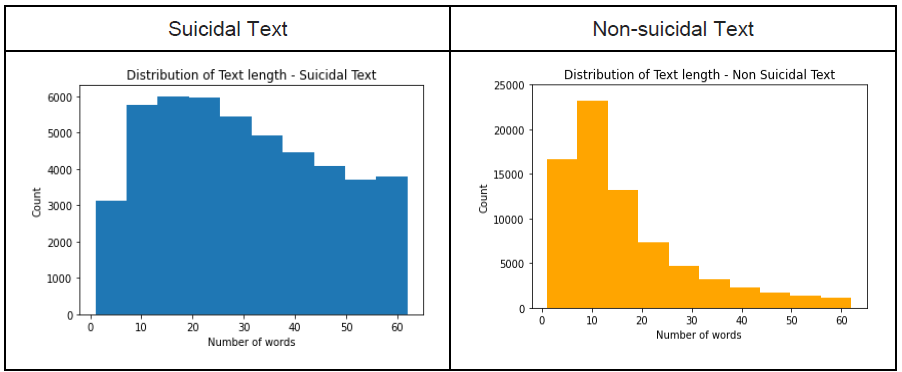
The dataset was first separated into three sets - training, testing, and validation - with a split ratio of 8:1:1. The train set was then explored to gain insight into the data, and each class was examined more closely.



**Fig. 3.6** Word Count & Word Cloud for suicidal text

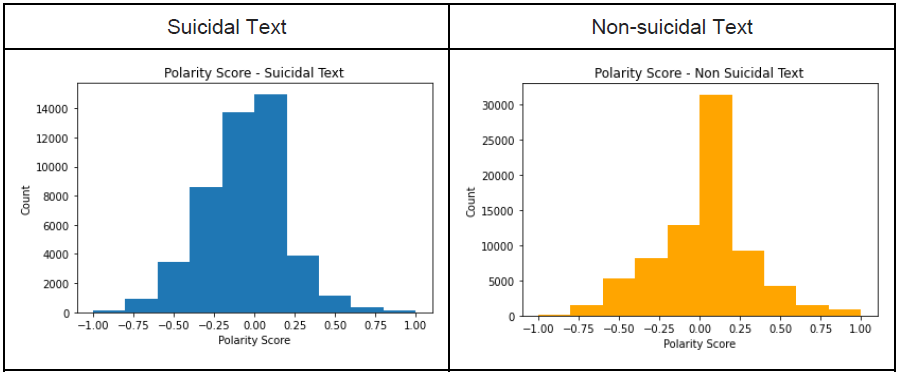
****

**Fig. 3.7** Word Count & Word Cloud for non-suicidal text



**Fig. 3.8** Distribution of text length for Suicidal (left) & non-Suicidal texts

The distribution of text length for the suicidal posts seems to be more uniform and higher in content in general as opposed to the text length of non-suicidal posts. This suggests that people with suicidal ideations are more expressive and vent about it on social media more than people with non-suicidal ideations.



**Fig. 3.9** Polarity Score Distribution for Suicidal & non-Suicidal Text

The polarity score from the content from suicidal data is more on the negative side, with, more data points in the negative region than from the texts with non-suicidal ideations that have a neutral reading into the polarity score, suggesting that texts identified with suicidal ideations do indeed have related data.

## CHAPTER 4

**REPRESENTATION LEARNING**

In natural language processing (NLP), representation learning refers to a collection of methods that transform unprocessed textual data into a computationally effective representation that can be utilized for machine learning tasks. NLP faces difficulties in achieving adequate performance due to the unstructured nature of natural language texts, which can vary greatly in terms of granularity, tasks, and domains. To address this, word embeddings are frequently employed to depict words as a condensed vector. This enables machine learning classifiers to identify other words with similar contexts and capture semantic and syntactic information of words. Advanced deep learning models such as Word2Vec, GloVe, and fastText are commonly employed to initiate the embedding layer, as their performance is typically superior to random initialization.

If pre-trained word embeddings are not loaded into the embedding layer, the model would need to learn the word embeddings from scratch during training, which could be challenging due to the sparsity of training data with numerous rare words that might not be accurately represented, as well as the longer training process resulting from a large number of trainable parameters.



**Fig. 4.1** Representations Built

Our objective is to generate representations utilizing both Word2Vec and GloVe algorithms. To accomplish this, we will pre-train customized Word2Vec embeddings using the training dataset, while simultaneously employing readily accessible GloVe embeddings.

## Word2vec Embeddings

## The use of Word2Vec embeddings can provide a compact vector representation of words and effectively capture information pertaining to their meaning. This is achieved by clustering together vectors of similar words and making estimations about the word's meaning based on its frequency in the text. These estimations can be utilized to establish word associations with other words present in the corpus.

## Word2Vec is a prediction technique based on neural networks that consists of two algorithms: continuous bag-of-words (CBOW) and skip-gram (SG). CBOW seeks to anticipate a target word by utilizing a set of context words, while SG takes the target word as input and aims to predict the context words that surround it both before and after.

## The Word2Vec algorithm comprises three primary components: (1) vocabulary builder, (2) context builder, and (3) neural network (Andrea, 2019). The vocabulary builder accepts raw sentence data and extracts distinct words to establish a corpus (Bhanawat, 2019). Subsequently, the context builder transforms the words into vectors by considering all words within the context window of the target word. In the end, Word2Vec employs a neural network with a single hidden layer for training, where the number of neurons in the layer corresponds to the dimensions of the embedding.

## Existing pre-trained word embeddings are often used for general language applications, or when the dataset is not large enough to generate customized embeddings that are meaningful. Considering that our training dataset's vocabulary size is around 20,400 with approximately 1,39,500 data points, it is suitable for a custom trained embedding layer to provide an optimal learning scenario.

## To train the Word2Vec embeddings, we have established certain parameters: a minimum count of 2, embedding dimension of 300, and context window of 10. The minimum count specifies that words that appear only once in the training data will be disregarded to avoid overfitting the trained embedding. Additionally, the embedding dimension has been set to a high value as studies have shown that vector representations' quality improves with a larger vector size. With a larger embedding size, words can be distinguished better as each dimension captures a particular aspect of the word's meaning.

## However, research indicates that the vector quality decreases after 300 dimensions; hence we have limited our embedding dimensions to 300. Lastly, the context window determines the number of words to be considered before and after the target word during the embedding training process. Typically, this value is set to around 5 or 10, and we have chosen the value 10.

## GloVe Embeddings

Global Vectors for Word Representation (GloVe) is an alternative unsupervised learning algorithm to acquire vector representation for words. Unlike Word2Vec, it is a count-based model rather than a predictive model and incorporates both local and global context by utilizing the word co-occurrence matrix.

GloVe uses a log-bilinear model. It combines the strengths of both count-based and predictive models, incorporating both local and global context by using a co-occurrence matrix. The model aims to minimize the difference between the dot product of two words' vectors and the logarithm of their probability of co-occurrences. The co-occurrence matrix is a large matrix that counts the number of times vocabulary items occur together within a context window across the entire corpus. The matrix is factorized to yield a lower-dimensional matrix by minimizing a "reconstruction loss". The ratio of probabilities derived from the matrix helps to distinguish relevant from irrelevant words and further differentiate between relevant words.

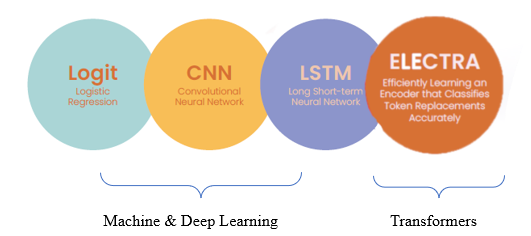
Using pre-trained GloVe word embeddings can indeed be beneficial for model performance, particularly in cases where the dataset is limited or similar to the data that was used to train the pre-trained embeddings. As you mentioned, the Twitter dataset used to train the 200-dimensions Twitter embeddings contains a similar type of data to your dataset, which makes it a good candidate for transfer learning. The larger dataset used to train the pre-trained embeddings can also result in better capture of word semantics and syntax, which can help improve your model's performance. However, it's worth noting that pre-trained embeddings may not always capture domain-specific nuances, so it's essential to evaluate the performance of the pre-trained embeddings on your task before deciding to use them in your model.

# CHAPTER 5

**MODEL BUILDING & EVALUATION**

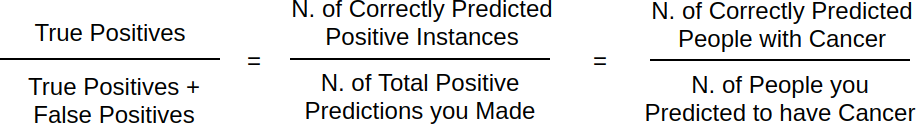
This section, we will build different models and evaluate their performance for classifying suicidal text from the Reddit dataset. Four models were constructed, including machine learning, deep learning, and transformer models. These models were Logistic Regression (Logit), Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Efficiently Learning on Encoder that Classifies Token Replacements Accurately (ELECTRA).

The following section describes the training, fine-tuning, and testing of various models for classifying suicidal text in our binary prediction problem. To evaluate the performance of the models, the test dataset will be used, and four metrics will be employed: Accuracy, Precision, Recall, and F1 score. The F1 score will be given greater consideration since it provides a well-balanced measure of model performance, considering both false positives and false negatives.

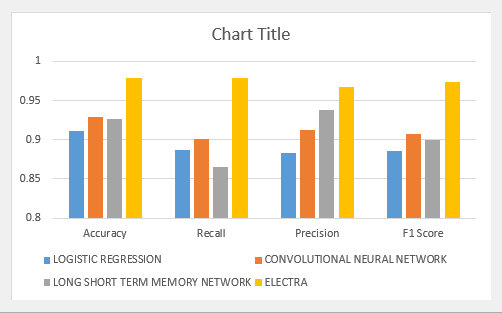


**Fig. 5.1** Range of Models built

A measure of precision (true positives) is the frequency with which accurate positive forecasts are generatedRecall is a metric that represents the proportion of positive cases in the data that the classifier correctly predicted out of all the positive cases. It is also known as sensitivity in some cases. The F1-Score is a metric that combines recall and precision. It is commonly referred to as the harmonic mean of the two. The harmonic mean is a different way of computing an "average" of numbers, and it is often more suitable for ratios than the traditional arithmetic mean, such as precision and recall.



**Fig.** **5.2** Precision Calculation Formula



**Fig. 5.3** Performance Analysis Graph of the Models

## Logit

## Logistic regression is a widely used technique for text classification, which is a type of supervised learning algorithm used to predict the likelihood of a categorical outcome. In text classification, the outcome variable is usually a binary variable that indicates whether a text document belongs to a particular class or not.

## The fundamental concept of logistic regression is to utilize a linear function to model the connection between the input features and the output variable. The linear function is then transformed using a sigmoid function to obtain a probability value between 0 and 1. This probability value indicates the likelihood that the input belongs to the positive class.

## In text classification, the input features are typically the words or phrases that appear in the text document. These features are represented as a vector of numerical values, such as word counts or TF-IDF scores. The output variable is a binary variable that represents whether the document belongs to a certain class or not.

## To train a logistic regression model for text classification, you would typically start by preprocessing the text data to extract the input features. This might involve tokenizing the text into individual words or phrases, removing stop words and punctuation, and performing stemming or lemmatization.

## Next, after splitting the data into training and testing sets, the logistic regression model is trained on the training data. During the training process, the model learns the optimal values of the coefficients of the linear function that best fit the input features to the output variable. Once the model is trained, it can be used to predict the class of new, unseen text documents based on their input features. The accuracy of the model's predictions can be evaluated on the testing data. Once the model is trained, you can use it to predict the probability of a new text document belonging to the positive class. You can also use the model to classify new documents into the positive or negative class based on a threshold probability value.

## We have two model variants with different settings for changes in the embedding layers. They are:

## Logit Model 1: Custom Word2Vec Embeddings (300 dimensions)

## Logit Model 2: Pre-trained GloVe Embeddings (200 dimensions)

## Default parameters were used for this model.

## The table below displays the performance of all variations of the Logistic Regression model. Model 1, which utilizes customized Word2Vec embeddings, has been determined to be the best-performing model and has outperformed Model 2, which uses pre-trained GloVe embeddings, across all the accuracy, precision, recall, F1 metrics that have been taken in consideration here.

## 

## Table 5.1. Logit Models Performance Comparison

## In prediction problems involving regression and classification, pre-trained word embeddings typically do not perform as well as creating new word embeddings from the original dataset. This could be due to the specific nature of suicide detection as a niche field, where there may be differences in the vocabulary and corpus used for training, rendering off-the-shelf embeddings less effective. Moreover, off-the-shelf word vectors may only be suitable for low-resource situations. However, our dataset is sufficiently large to develop meaningful representations using custom embeddings, and creating custom-trained word embeddings would be more effective, particularly for our niche application.

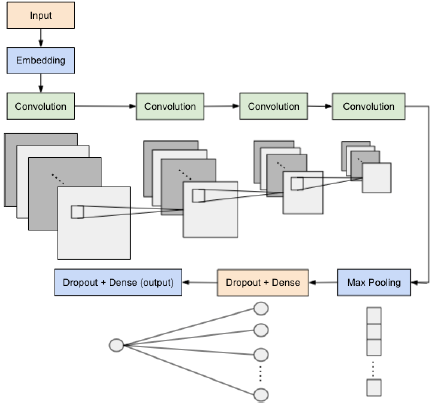
## Architecture of a Logistic Regression Model [56]. | Download Scientific Diagram

## Fig. 5.4 Logistic Regression Architecture

## CNN

CNNs have also been successfully used for text classification tasks. CNNs were originally developed for image processing, but they have been adapted for text classification by treating the text as a one-dimensional sequence of data.

CNNs are a type of deep learning model commonly used in image and text classification tasks. In the case of text classification, the input data is usually in the form of word embeddings or sequences of characters. In a CNN for text classification, the input data is first represented as a matrix, where the rows represent the words or characters in the input sequence, and the columns represent the dimensions of the word embeddings. The matrix is then passed through a convolutional layer, which applies a set of filters to the input matrix. The filters, which are learnable parameters of the model, slide over the matrix and perform element-wise multiplication and summation operations, producing a feature map. The feature map is then passed through a non-linear activation function, such as ReLU or sigmoid, and then through a pooling layer, which reduces the dimensionality of the output by either taking the maximum or the average of a local region of the feature map. The output of the pooling layer is then passed through one or more fully connected layers, which perform a linear transformation on the output and apply a non-linear activation function. Lastly, the output from the final fully connected layer is fed through a softmax activation function to obtain a probability distribution across the output classes.



**Fig. 5.5** CNN Model Architecture

The convolutional layers use filters of different sizes to extract features from the input text data. Each filter slides over the embedded data and performs a convolution operation to produce a new set of feature maps. The pooling layer then applies max-pooling to each feature map to reduce the output dimensionality. The resulting pooled features are then passed to a dropout layer to prevent overfitting during training. Finally, a fully connected layer is used to classify the input data into the two classes of suicidal and non-suicidal. The output of this layer is then fed through a softmax activation function, which generates a probability distribution across the output classes. The purpose of the embedding layer is to represent words in a semantically meaningful space, where words with similar meanings are closer together, while words with different meanings are farther apart (Kim and Jeong, 2019). Next, the convolutional layers are employed to extract contextual features from the input data.

The earlier convolutional layers extract simple contextual information, while the later ones capture more complex features and extract sentiments that affect classification. Following the convolutional layers, we applied max-pooling, which allows our model to focus on the most important features in the sentence, regardless of word order (Goldberg, 2015). To prevent overfitting, we included a dropout layer as a form of regularization in our CNN model. Finally, we used a fully connected layer to connect all input and output neurons together. The result of this layer undergoes a sigmoid activation function to determine whether the text is categorized as suicidal or non-suicidal. We have conducted experiments using several model variants, each with variations made to the Embedding layer. They are:

* CNN Model 1: Random Initialization (No pre-trained weights)
* CNN Model 2: Custom Word2vec Embeddings (300 dimensions)
* CNN Model 3: Pre-trained GloVe Embeddings (200 dimensions)

We conducted experiments to determine the optimal hyperparameter values empirically and arrived at the following values: an embedding size of 300 for model(s) 1 and 2, and an embedding size of 200 for model 3. The difference in embedding size was due to the dimension of the loaded embedding weights into the embedding layer. We utilized four convolutional layers, with 32 filters within each layer.

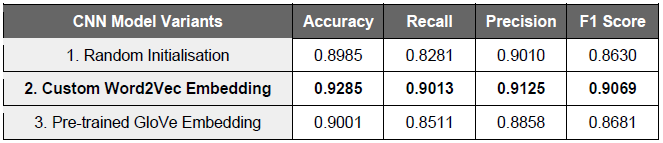
We applied filter sizes of 5x5, 6x6, 7x7, and 8x8 in convolutional layers 1, 2, 3, and 4, respectively. Additionally, we applied a dropout rate of 0.2 to prevent overfitting. We utilized the Adam optimizer with a learning rate of 0.00001, and set our loss function to be Binary Cross Entropy with Logits Loss (BCEWithLogitsLoss), which combines Sigmoid and Binary Cross Entropy Loss (BCELoss). This loss function provides numerical stability using the log-sum-exp trick, which is not achievable when implementing the plain Sigmoid followed by BCELoss functions sequentially. Ultimately, we opted to use the (ReLU) as the activation function in each convolutional layer to prevent the exponential expansion of computation necessary in our neural network.



**Fig. 5.6** CNN Model Training Graph

The figure above displays a graph that illustrates the accuracy of training and validation for all variations of the CNN model over the course of several epochs. The graph shows that as the number of epochs increases, the accuracy of both the training and validation sets begins to level off, indicating that the model training process was stable for all variations of the model.

The table presented below shows the performance of all variations of the CNN model. The top-performing model variant is Model 2 (Custom Word2Vec Embeddings), which has achieved the best results across all performance metrics. This result is consistent with the findings of the Logistic Regression model, which also showed that custom Word2Vec embeddings performed the best.

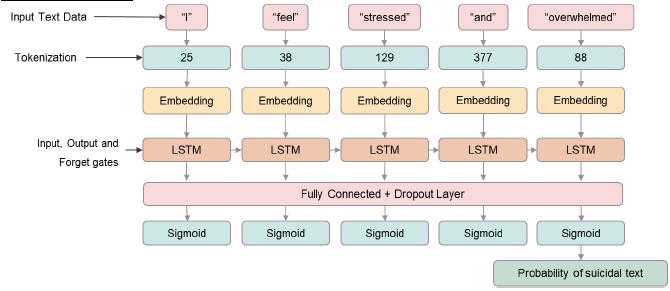


**Table 5.2.** CNN Models Performance Comparison

**5.3. Long Short Term Memory (LSTM)**

LSTM is a type of RNN architecture widely used in text classification tasks. Unlike traditional RNNs, which suffer from the vanishing gradient problem, LSTM is able to remember information for longer periods of time by using a gating mechanism to selectively forget or retain information.

The output of each LSTM cell is determined by the input at the current time step, the previous hidden state, and the previous cell state. The previous hidden state and cell state are updated at each time step based on the current input and the gates' decisions, allowing the LSTM to model long-term dependencies in the input sequence. LSTM models can be used for text classification by taking in a sequence of words and producing a single output indicating the classification of the text. LSTM models have proven to be very effective in modeling sequences and have been used extensively in natural language processing tasks. The ability of LSTM cells to selectively retain or forget information over long sequences makes them especially well-suited for text classification tasks where the order of words in a sentence is important.

****

**Fig. 5.7** LSTM Model Architecture

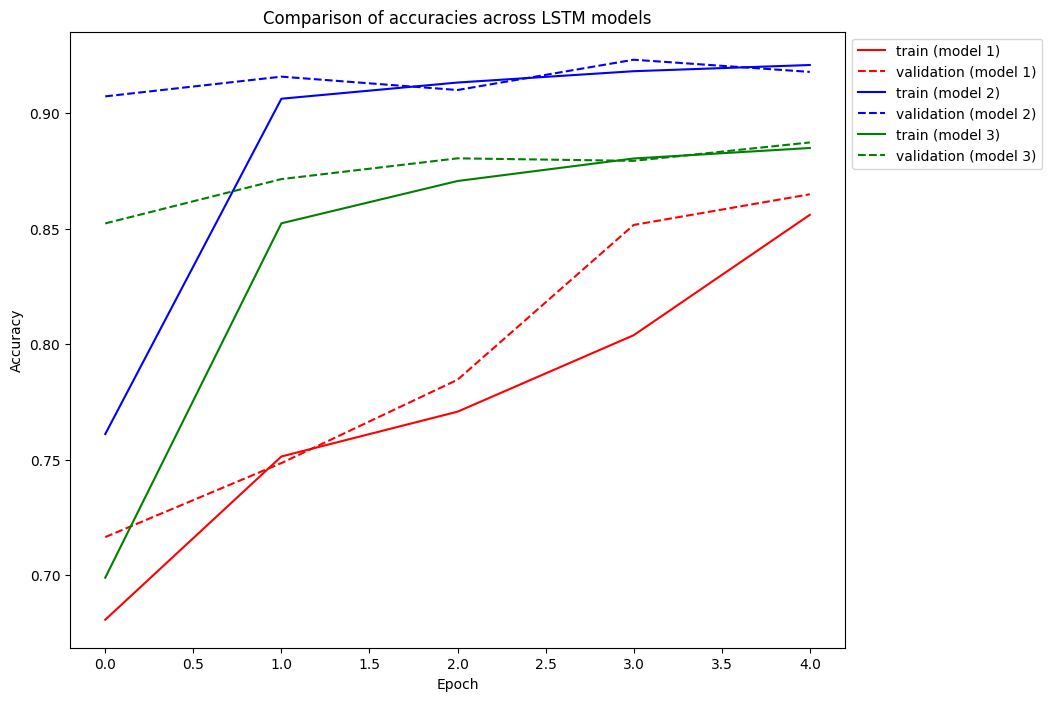
The five-layer LSTM model begins with word tokenization, followed by an embedding layer that converts integer tokens to embedding vectors. The LSTM layer handles the input data, making decisions on what information to retain, discard, or output. Following that, the third layer is a fully connected layer that maps the LSTM's output to a size of 1, while the fourth layer applies the sigmoid activation function to classify the input as either suicidal or non-suicidal. The final layer reshapes the output to match the batch size.

This architecture is illustrated in the figure above.

We have conducted experiments using several model variants, each with variations made to the Embedding layer. They are:

* LSTM Model 1: Random Initialization (No-pre-trained weights)
* LSTM Model 2: Custom Word2vec Embeddings (300 dimensions)
* LSTM Model 3: Pre-trained GloVe Embeddings (200 dimensions)

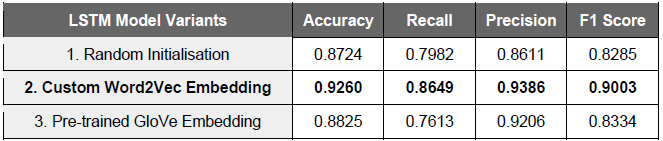
Just like with the CNN model, we conducted experiments with various combinations of hyperparameters to determine the best values. Our final choice for the LSTM model includes an embedding size of 300, 128 hidden dimensions, and 2 layers within the LSTM layer. We also set the dropout rate to 0.5 and used Adam as the optimizer with a learning rate of 0.00001, along with the BCEWithLogitsLoss loss function.



**Fig 5.8** Long Short Term Memory (LSTM) Model Training Graph

The figure above shows the training graph for all LSTM model variations of train and validation accuracies across epochs. Similar to CNN, the accuracies have started to plateau as the number of epochs increases, indicating that the model training process was stable for all model variants.

Table 4 presents the model performance for all LSTM model variants, with the best model being Model 2 (Custom Word2Vec Embeddings). This model has outperformed other models across all metrics, which is consistent with the performance of the Logistic Regression and CNN models where custom Word2Vec embeddings also performed the best.

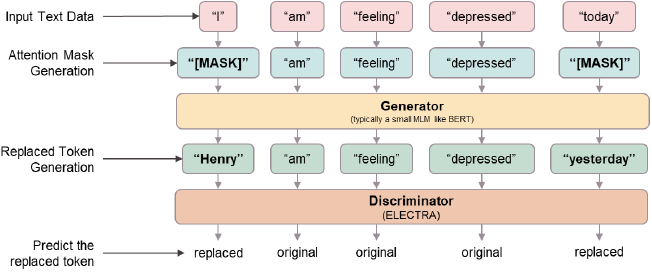


**Table 5.3.** LSTM Models Performance Comparison

**5.4. Transformers**

The transformer model is an architecture that utilizes attention-based encoder-decoder structure and self-attention mechanism to improve word representations in NLP tasks. Compared to deep learning models such as CNN and LSTM, transformers are capable of handling long-range interactions more effectively and do not suffer from the vanishing gradient problem. Compared to conventional models, transformers can be parallelized because they don't rely on recursive operations. They process entire sentences at once using attention mechanism and positional embeddings, enabling them to handle long-range interactions more effectively than models like CNN and LSTM, which suffer from vanishing gradient problems. Typically, transformer models are first pre-trained on large datasets using unsupervised methods and then fine-tuned for specific prediction tasks using supervised techniques, resulting in significantly improved performance and reduced training time. In this segment, we will optimize the most recent transformer model called ELECTRA for the detection of suicidal text. ELECTRA is a transformer model that was pre-trained by Google in 2020. It surpasses other advanced transformer models like RoBERTa and XLNet in terms of performance on benchmark datasets, while demanding very less computational power. Unlike RoBERTa and XLNet, ELECTRA achieves superior results with a smaller dataset and shorter training durations.

In contrast to BERT, which uses the MLM pre-training task, ELECTRA introduced the RTD pre-training task to overcome the limitations of MLM. In MLM, a small subset of tokens in the input are masked and the model is trained to predict those tokens. Nonetheless, this strategy causes the model to solely focus on predicting a limited segment of the input, which ultimately reduces the amount of knowledge acquired from each sentence. Although BERT still demonstrates good performance in subsequent NLP tasks, it necessitates substantial computational resources. ELECTRA's RTD task addresses these issues and has been shown to outperform BERT-based models on certain benchmark datasets with significantly less computational power (Clark et al., 2020).



**Fig. 5.9** ELECTRA RTD System

ELECTRA is different from BERT in that it replaces input tokens with incorrect but plausible fake tokens instead of masking them. This is done through a pre-training task called RTD, which trains a multi directional model to identify replaced tokens using a discriminator. The replaced tokens are generated from a small MLM like BERT and are trained jointly with the discriminator while sharing the same input word embeddings. ELECTRA's ability to learn from accurate data representation makes it more efficient than BERT, and it requires less training data to achieve the same model performance. The discriminator model is available in three variants: small, base, and large. The small and base models are pre-trained using BERT data, while the large model is pre-trained using XLNet data.

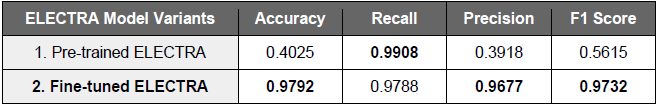
For implementing the ELECTRA model, the base model is used through the Hugging Face Transformers, similar to BERT. The base model is chosen as it has a similar size to BERT, which allows for a better comparison between the performance of the two transformer models. The original data is passed through the Hugging Face ELECTRA tokenizer for preprocessing.

We have experimented the ELECTRA Model with the following variants in fine tuning:

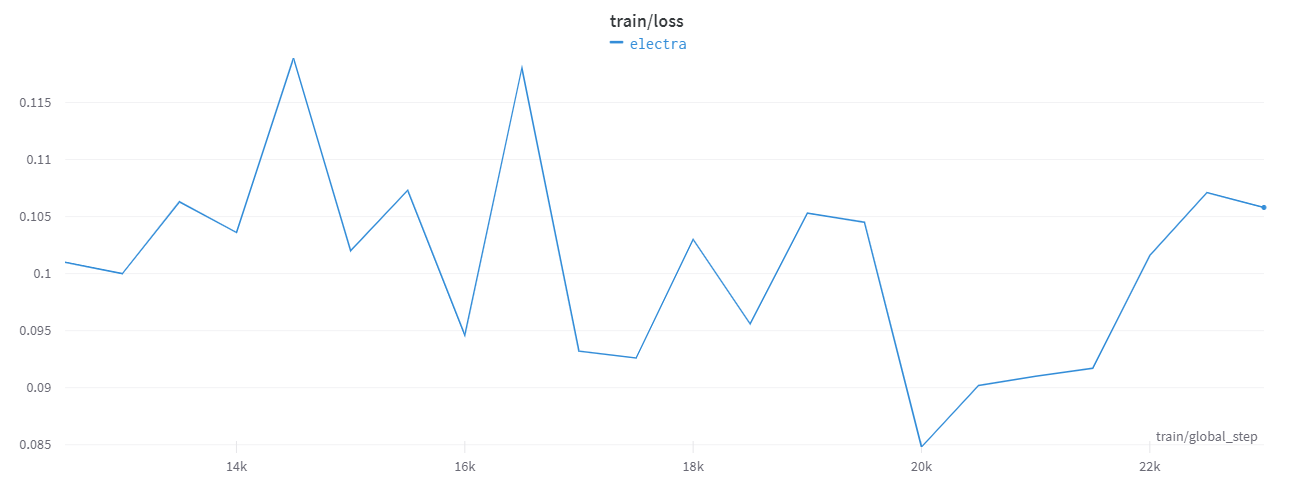
* ELECTRA Model 1: Pre-trained
* ELECTRA Model 2: Fine-tuned

We fine-tuned our model for 1 epoch of batch size of 6 and a light learning rate of 0.00001. Typically, it is recommended to fine-tune for 1 or 2 epochs to avoid overfitting. Given the large dataset size, 1 epoch should be sufficient. Although a larger batch size can speed up training, the memory constraint due to the long sequence text forced us to use a small batch size. Finally, we intentionally set a small learning rate to avoid overfitting and to incrementally adapt the pretrained weights. A large learning rate can potentially destroy the pre-trained features with large gradient updates during fine-tuning.

The table below shows the performance of all ELECTRA model variants. Model 2 (Fine-tuned ELECTRA) performed the best and had significantly better accuracy, precision, and F1 score compared to Model 1 (Pre-trained ELECTRA). However, Model 1 had a higher recall score, which was close to 1. Nevertheless, our model selection was based on the F1 score, so the higher recall score of Model 1 did not affect our choice.



**Table 5.4.** ELECTRA Models Performance Comparison

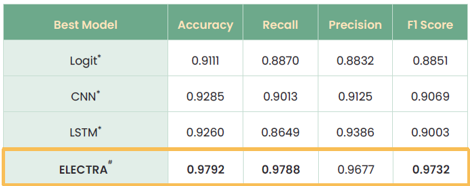


**Fig. 5.10** ELECTRA Run Train/Loss Graph

Figure above shows the training graph for fine-tuned ELECTRA of training loss over 1 epoch, with 500 data points representing 1 step. We could see a decreasing trend in the training. At the end of the training, ELECTRA had a significantly less loss. As the training loss has started to plateau and stabilize at the end of the epoch, we can conclude that the training on 1 epoch was adequate.

**5.5. Model Selection**

Table below provides an overview of the main outcomes of the models implemented. The best results in Logit, CNN, and LSTM were obtained using custom trained Word2Vec embeddings, while for ELECTRA, fine-tuning on our dataset gave the best results.



**Table 5.5.** AllModels Performance Comparison

ELECTRA has achieved the highest F1 score, accuracy, and recall score, while CNN achieved the second highest F1 score. Although CNN's performance is comparable to ELECTRA, the latter has demonstrated superior performance and speed due to its RTD approach, which directly addresses previous models' limitations. Hence, ELECTRA will be the final model selection for this task.

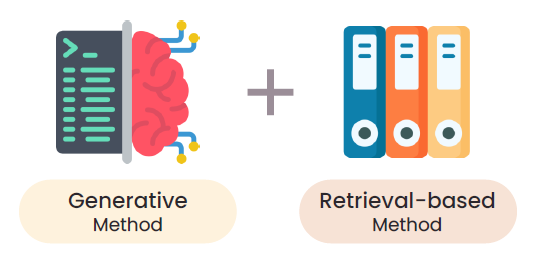
# CHAPTER 6

**CHATBOT INTEGRATION**

**6.1. Implementation**

Our chatbot utilizes transformers, which have a self-attention mechanism that can enhance performance compared to traditional recurrent methods. The generative aspect of the chatbot is powered by Dialo-GPT, a pre-trained transformer-based model designed for multi-turn conversations and was created by Microsoft in the year of 2019. Dialo-GPT was built on 147 million multi-turn dialogue from Reddit discussion threads and is available in three variations, namely small, medium, and large, similar to other transformer models. We utilized the original pre-trained Dialo-GPT model from the Hugging Face Transformers library.

Although we attempted to train a customized chatbot, we needed a large chat dataset and training resources to generate coherent and relevant responses. Nevertheless, by a single-turn conversation Turing test conducted by Zhang et al. (2019), responses generated by Dialo-GPT were comparable to human response quality. It should be noted that the pre-trained Dialo-GPT chatbot is not equipped to provide appropriate replies to suicidal messages. The chatbot now has a retrieval-based component that searches for comforting messages from a library of various suicidal prevention websites and a list of local helplines for immediate support when users input signals suicidal intent. This feature allows the chatbot to respond appropriately by providing comforting messages and helpful information.

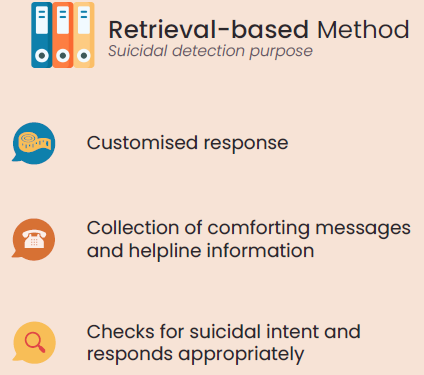


**Fig.6.1** Neural Response Generation (NRG) Model

Chatbots are models that generate responses based on input text and can be classified into two main types: neural response generation (NRG) models. The first type is retrieval-based, which selects a response from a set of pre-existing responses using certain rules. The second type is generative, which uses machine learning to create responses from scratch. By combining generative methods and retrieval-based methods, the chatbot will be able to provide a more comprehensive and personalized conversation with users while also being able to handle potentially sensitive situations such as suicidal messages.

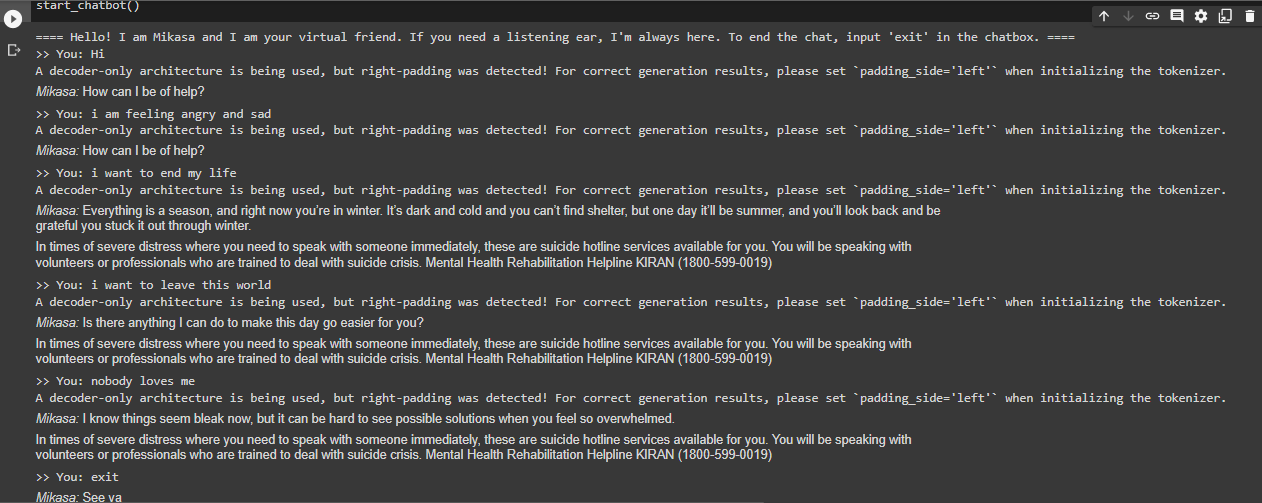


**Fig.6.2** Reasons to use Dialo-GPT

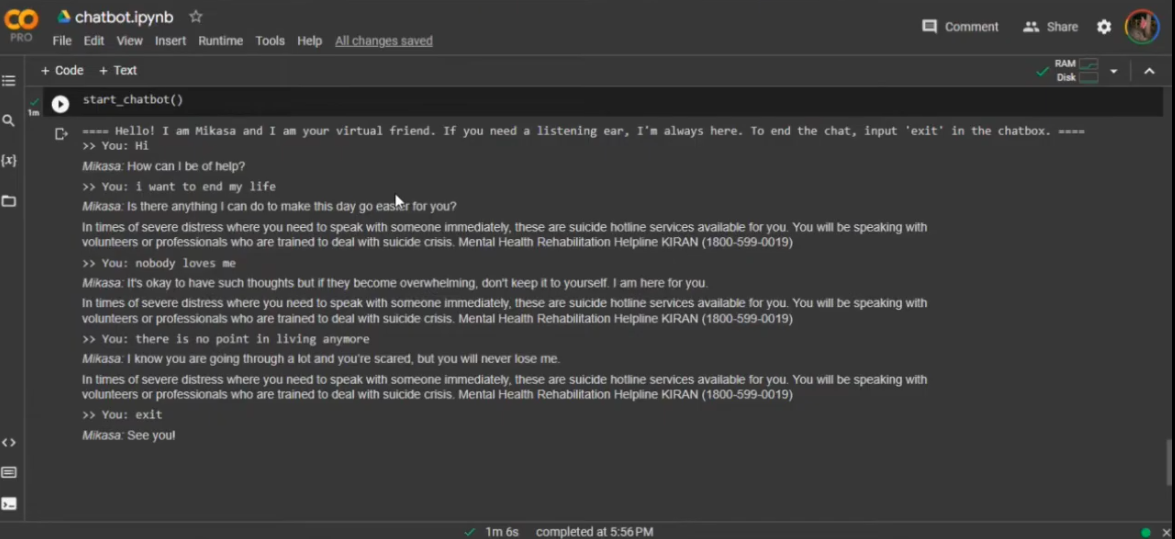


**Fig. 6.3** Reasons for Retrieval-based Method

Our chatbot was created using transformers, as the self-attention mechanism has been shown to perform better than traditional recurrent methods. DialoGPT was trained on 147 million multi-turn posts from Reddit communication threads and is available in three variations (small, medium, and large). We used the original pre-trained DialoGPT model from the Hugging Face Transformers library, as attempts to train a custom chatbot with a smaller dataset were unsuccessful in producing comprehensible and relevant responses. DialoGPT's responses were found to be comparable to human responses in a single-turn conversation Turing test conducted by Zhang et al. (2019). However, as DialoGPT was not trained to provide suitable responses to suicidal messages, we added a retrieval-based component to the chatbot. We compiled a library of comforting messages and a list of local helplines from various suicidal prevention websites. One way to check for suicidal intent is to use natural language processing techniques to analyze the user's input and identify any concerning patterns or keywords. Once a message is flagged as potentially suicidal, the chatbot can retrieve a pre-written response from its library of comforting messages and local helplines.



**Fig. 6.4** Chatbot Demo #1



**Fig. 6.5** Chatbot Demo #2

The image presented in Figures 6.1 & 6.2 depict an illustration of our mental health chatbot, which is currently available for use through our Google Colab notebook script.

The developed chatbot accurately works on the neural response generation (NRG) model to deliver the desired output.

**6.2. Limitations**

## We agree that chatbots can be a valuable tool in the mental health, particularly in providing support and resources to individuals who may not have access to traditional mental health services. However, as you mentioned, there are some limitations to chatbots' ability to give back empathetic feedback and accurately identify suicidal risks.

## Chatbots at universities - these are the pros and cons

## Fig. 6.6 Disadvantages of a chatbot

## No Internet Vector Art, Icons, and Graphics for Free Download

## Fig. 6.7 Necessity of simultaneous internet connection

## Therefore, it is important to continue researching and developing chatbots that can provide more nuanced and effective answers. Additionally, ensuring the privacy and security of personal health information is crucial for the widespread adoption of mental health chatbots. Overall, chatbots have the potential to be a valuable complement to traditional mental health services, but careful consideration and development are necessary to ensure their effectiveness and ethical use.

## 6.3. Conclusion

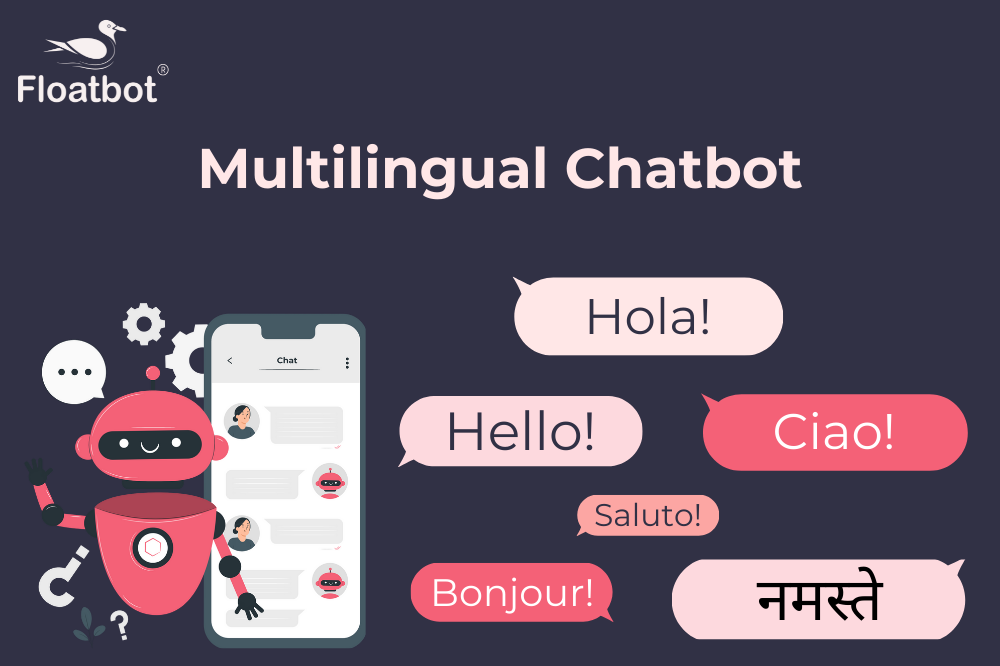
Through ELECTRA transformer technology, we have likely created a chatbot capable of engaging in natural and fluid conversations with users while providing valuable mental health support and resources. Overall, our research highlights the potential of modern machine learning techniques to support mental healthcare and improve people's quality of life. By leveraging the power of ELECTRA transformer technology, you have demonstrated how advanced natural language processing capabilities can be used to develop innovative and effective mental health tools. It is hoped that our work will inspire further research in this area and contribute to the development of more sophisticated mental healthcare chatbots in the future and support individuals who may be struggling with mental health issues. Addressing mental health issues and providing timely support to struggling people is essential.



**Fig. 6.8** Chatbot in action help make a phone call to the helpline number provided in the system

**6.4. Future Improvements**

* Building a multilingual chatbot:



**Fig. 6.9** Depiction of a multilingual chatbot

Based on a study conducted by Cameron et al. (2019) on the usability of mental health care chatbots, users suggested that the ability to input different languages would enhance interaction with the chatbot. Our current chatbot only supports English conversations, so incorporating additional languages could increase our audience to a more diverse international community. This could be accomplished by training and fine-tuning our models on conversational data in other languages, which could be sourced by scraping social media platforms. Alternatively, we could use existing models from the online community, such as the DiGPTame model (Cooper, n.d.), which is a fine-tuned version of DialoGPT for Spanish language data. However, this improvement may only be feasible for more widely spoken languages, as acquiring sufficient conversational data for dialects can be challenging.

* Integration of chatbot onto social media platforms:

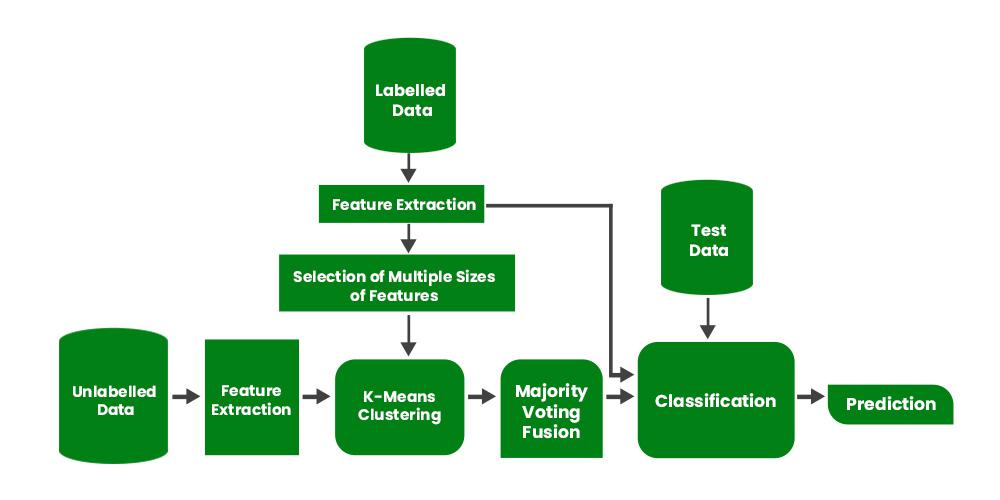


**Fig. 6.10** Animated depiction of a chatbot on multiple platforms

According to a report by Statista Research Department (2021), social media is one of the most popular online activities and is expected to have 4.41 billion users by 2025. Social media has also been found to be a source of social support for people feeling isolated and lonely during the COVID-19 pandemic. Given the increase in social media usage, researchers have suggested that social media platforms can be used to detect mental health symptoms (Torous et al. 2021). Therefore, it is possible to integrate our chatbot onto a popular social media platform to increase its usage and visibility.

Other than Reddit, Facebook is one such popular social media platform, with about 2.91 billion monthly active users (Statista Research Department, 2021). Facebook has also attempted to identify content related to suicide, which is then communicated to law enforcement teams. However, this has raised concerns over privacy standards, as Facebook has not obtained user consent to disclose such information to a third party (Goggin, 2019). In contrast, our chatbot proposal is different as it does not involve a third party managing mental health data. Furthermore, involving a third party could result in longer waiting times before help can be provided, whereas our chatbot can reach out to the account holder on Facebook immediately. Therefore, we propose integrating our chatbot onto Facebook to provide timely assistance and address privacy concerns.

* Improve Data Quality by Adopting Semi-Supervised Learning:



**Fig. 6.11** General architecture diagram of semi-supervised learning

To enhance the accuracy of data labeling, semi-supervised learning techniques like pseudo-labeling can be employed. By using a limited set of human-annotated data, deep neural networks can be trained in a supervised manner. Subsequently, the trained model can generate labels for unlabeled data, and a combination of both human-labeled and pseudo-labeled data can be utilized for further training. This approach helps improve the quality of data utilized for training the models, thereby enhancing their robustness. While pseudo-labeled data may not be as precise as human-annotated data, it significantly reduces the amount of manual effort and time required for labeling.

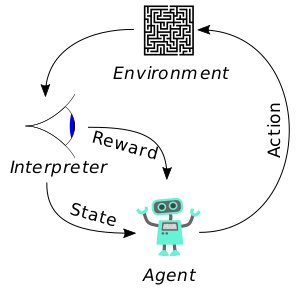
* Larger Transformer Model Assembly to Improve Model Performance:



**Fig. 6.12** Overview of large language models

The most outstanding performing model we currently have is the ELECTRA model, which was fine-tuned on our data. Unfortunately, we were unable to incorporate the larger ELECTRA model due to computational limitations. Generally, larger transformer models, which are pre-trained with more data, layers, and parameters, have been shown to perform better on benchmark datasets such as GLUE10 (Clark et al., 2020). Although utilizing larger transformer models may improve our model’s performance, it is important to consider that they require longer training time and may be more prone to overfitting on the training dataset.

* Reinforcement Learning to Improve chatbot Response:



**Fig. 6.13** Simplified depiction of how reinforcement learning works

Reinforcement learning (RL) is a type of machine learning that allows an agent to learn how to behave in an environment by trial and error. In the context of chatbots, RL can be used to improve the chatbot's response to user input.

Here are the steps involved in using RL to improve chatbot response:

1. Define the environment. The environment is the chatbot's interaction with the user. This includes the user's input, the chatbot's response, and the user's feedback.
2. Define the agent. The agent is the chatbot. The agent's goal is to learn how to generate responses that are satisfactory to the user.
3. Define the reward function. The reward function is a way of measuring how good the agent's response is. The reward function should be designed to encourage the agent to generate responses that are satisfactory to the user.
4. Initialize the agent's policy. The agent's policy is a function that maps from the environment to the agent's response. The policy can be initialized randomly or by using a human expert.
5. Repeat the following steps:
   * The agent takes an action.
   * The environment responds to the agent's action.
   * The agent receives a reward.
   * The agent updates its policy based on the reward.

This process is repeated until the agent learns how to generate responses that are satisfactory to the user.

RL has been shown to be an effective way to improve chatbot response. In one study, a chatbot that was trained using RL was able to generate more satisfactory responses to user input than a chatbot that was trained using traditional machine learning techniques.

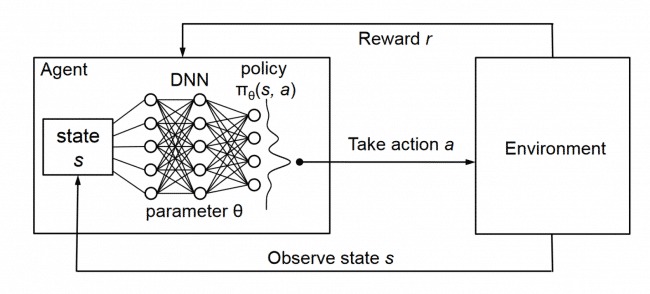
Some of the advantages of using RL to improve chatbot response are:

* RL can be used to learn complex tasks that are difficult or impossible to learn using traditional machine learning techniques.
* RL can be used to learn from experience. This means that the chatbot can improve its response over time as it interacts with more users.
* RL can be used to learn in a dynamic environment. This means that the chatbot can adapt its response to changes in the user's input or the environment.

Some of the challenges of using RL to improve chatbot response are:

* RL can be computationally expensive.
* RL can be difficult to train.
* RL can be sensitive to the choice of reward function.

Despite the challenges, RL is a promising approach to improving chatbot response. As RL technology continues to develop, it is likely to become an even more effective way to train chatbots that can provide a high level of customer service.



**Fig. 6.14** Architecture of Reinforcement Learning

Reinforcement learning can potentially be integrated into our chatbot to consistently improve its capabilities and responses. This method allows the chatbot to learn by interacting with its environment, which in this case is the end users. Reinforcement learning can be used to train a chatbot by providing a reward signal based on the user's feedback. The chatbot can learn from this feedback and adjust its behavior to maximize the reward. In the case of a mental health chatbot, the reward can be based on the user's level of satisfaction or perceived helpfulness of the chatbot's responses. Over time, the chatbot can learn from its interactions and improve its ability to generate appropriate responses and provide effective support to those in need. However, this learning process requires an abundance of user conversations, which can take a long time to achieve favorable performance.

# REFERENCES

1. Arya, Vanshika & Khan, Rukhsar & Aggarwal, Prof. (2022). A Chatbot Application by using Natural Language Processing and Artificial Intelligence Markup Language. International Journal of Soft Computing and Engineering. 12. 1-7. 10.35940/ijsce.C3566.0712322.
2. Tirumala, Kushal & Markosyan, Aram & Zettlemoyer, Luke & Aghajanyan, Armen. (2022). Memorization Without Overfitting: Analyzing the Training Dynamics of Large Language Models. 10.48550/arXiv.2205.10770.
3. Zhang, Xiaopeng & Qin, Liangxi. (2022). An Improved Extreme Learning Machine for Imbalanced Data Classification. IEEE Access. PP. 1-1. 10.1109/ACCESS.2022.3142724.
4. Rahali, Abir & Akhloufi, Moulay. (2023). End-to-End Transformer-Based Models in Textual-Based NLP. AI. 4. 54-110. 10.3390/ai4010004.
5. Gabín, Jorge & Ares, M. & Parapar, Javier. (2023). Keyword Embeddings for Query Suggestion. 10.48550/arXiv.2301.08006.
6. Bao, Han-Wu-Shuang & Wang, Zi-Xi & Cheng, Xi & Su, Zhan & Yang, Ying & Zhang, Guang-Yao & Wang, Bo & Cai, Huajian. (2023). Using word embeddings to investigate human psychology: Methods and applications. 31. 887. 10.3724/SP.J.1042.2023.00887.
7. Ricciardelli, Elena & Biswas, Debmalya. (2019). Self-improving Chatbots based on Reinforcement Learning.
8. Cameron, Gillian & Cameron, David & Megaw, Gavin & Bond, Raymond & Mulvenna, Maurice & O ' Neill, Siobhan & Armour, Cherie & Mctear, Michael. (2019). Assessing the Usability of a Chatbot for Mental Health Care. 10.1007/978-3-030-17705-8\_11.
9. Ce, Peng & Tie, Bao. (2020). An Analysis Method for Interpretability of CNN Text Classification Model. Future Internet. 12. 228. 10.3390/fi12120228.
10. Barbosa Pina, Débora & Kunstmann, Liliane & Bevilaqua, Felipe & Siqueira, Isabela & Lyra, Alan & de Oliveira, Daniel & Mattoso, Marta. (2022). Capturing Provenance from Deep Learning Applications Using Keras-Prov and Colab: A Practical Approach. Journal of Information and Data Management. 13. 10.5753/jidm.2022.2544.
11. Ni, Shiwen & Kao, Hung-Yu. (2022). ELECTRA is a Zero-Shot Learner, Too. 10.48550/arXiv.2207.08141.
12. Viduani, Anna & Cosenza, Victor & Araújo, Ricardo & Kieling, Christian. (2023). Chatbots in the Field of Mental Health: Challenges and Opportunities. 10.1007/978-3-031-10698-9\_8.

# CODE & IMPLEMENTATION

# (Data cleaning, data pre-processing, CNN & ELECTRA models, chatbot integration)

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# ’

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

|  |  |  |
| --- | --- | --- |
| **SRM** **INSTITUTE** **OF** **SCIENCE** **AND** **TECHNOLOGY**  **(Deemed** **to** **be** **University** **u/s** **3** **of** **UGC** **Act,** **1956)** | | |
| **Office** **of** **Controller** **of** **Examinations** | | |
| REPORT FOR PLAGIARISM CHECK ON THE SYNOPSIS/THESIS/DISSERTATION/PROJECT REPORTS | | |
| 1 | Name of the Candidate  **(IN** **BLOCK** **LETTERS)** | **DHRUV KANT LADIA**  **SAATVIK SHUKLA** |
| 2 | Address of the Candidate | FE-371, Near Tank-12, Sector-3, Salt Lake City, Kolkata - 700106,  West Bengal, India    1708, Chaurang CHS, Sector 16, Sanpada, Navi Mumbai, Mumbai-  400705, Maharashtra, India |
| 3 | Registration Number | 1. RA1911003010550  2. RA1911003010520 |
| 4 | Date of Birth | 27-11-1999  22-03-2001 |
| 5 | Department | Dept. of Computing Technologies |
| 6 | Faculty | Dr. R. Yamini |
| 7 | Title of the Synopsis/ Thesis/ Dissertation/Project | Identifying Suicidal Ideation: A Detection & Prevention System |
| 8 | Whether the above project /dissertation is done by | ~~Individual~~ or group:   1. If the project/ dissertation is done in group, then how many students together   completed the project : 2 (Two)   1. Mention the Name & Register number of other candidates :   Saatvik Shukla, RA1911003010520 |
| 9 | Name and address of the Supervisor / Guide | Dr. C. Pretty Diana Cyril  Assistant Professor, Department of Computing Technologies,  SRM Institute of Science and Technology, Kattankulathur-603203  **Mail ID:** [prettydc@srmist.edu.in](mailto:prettydc@srmist.edu.in)  Mobile Number: +919790086808 |
| 10 | Name and address of the Co-Supervisor / Co- Guide (if any) | NIL |
| 11 | Software Used | Turnitin |
| 12 | Date of Verification | 14/05/2023 |

**PLAGIARISM** **REPORT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Plagiarism** **Details:** **(to** **attach** **the** **final** **report)** | | | |
| **Chapter** | **Title** **of** **the** **Chapter** | **Percentage** **of** **similarity** **index** **(including** **self**-**citation)** | **Percentage** **of** **similarity** **index** **(Excluding** **self** **citation)** | **%** **of** **plagiarism** **after** **excluding** **Quotes,** **Bibliography,** **etc.,** |
| **1** | **Introduction** | **3** | **3** | **3** |
| **2** | **Literature** **Survey** | **2** | **2** | **2** |
| **3** | **Data Collection & Wrangling** | **<1** | **<1** | **<1** |
| **4** | **Representation Learning** | **1** | **1** | **1** |
| **5** | **Model Building & Evaluation** | **2** | **2** | **2** |
| **6** | **Chatbot Integration** | **<1** | **<1** | **<1** |
| I / We declare that the above information has been verified and found true to the best of my / our knowledge. | | | | |
| **Signature** **of** **the** **Candidates** | | **Signature** **of** **the** **Staff who checks plagiarism** | | |
| **Signature** **of** **the** **Supervisor/Guide** | | **Signature** **of** **the** **HOD** | | |