Predicting Suicidal Intent from Social Media Text Post Using Machine Learning

by

Md. Mubin Ul Islam Chowdhury
18101640
Mehadi Hasan
18101657
A.K.M Muhibullah Nayem
18301117
Humaira Tasnim Meem
17201103

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
May 2022

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md. Mubin Ul Islam Chowdhury

18101640

A.K.M Muhibullah Nayem

18301117

Mehadi Hasan

18101657

Humaira Tasnim Meem

17201103

Approval

The thesis/project titled "Predicting Suicidal Intent from Social Media Text Post Using Machine Learning" submitted by

1. Md. Mubin Ul Islam Chowdhury (18101640)
2. Mehadi Hasan (18101657)
3. A.K.M Muhibullah Nayem (18301117)
4. Humaira Tasnim Meem (17201103)

Of Spring, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May,2022.

Examining Committee:

Supervisor:
(Member)

Hossain Arif

Assistant Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr Md Golam Rabiul Alam

Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

1) This material is our own unique work, which has not been recently published somewhere else.
2) The paper is not lately being considered for publication somewhere else.
3) The paper reflects our own exploration and examination in an honest and complete way.

Abstract

We are living in an age of modern science where cutting-edge technology has made the world so small. We can now easily connect with people worldwide via the internet and using social media. Social media has become a popular way to connect with people and to share our thoughts and feeling with the people we are connected. People are using social media as a tool where they share their feelings, daily life activities and so on. As a result, people are spending more times on those platforms to connect with people rather than in person. People who suffer from suicidal ideation are expressing their feelings and emotions on social platforms. As suicide is now an alarming problem in our society, we can use machine learning technology to determine suicidal ideation in the early stage based on social media data such as Twitter data and Reddit data. We have combined deep learning and an artificial neural network to make a model that we have named SIP (Suicidal Intent Prediction) which can detect suicidal ideation based on the text data of social media in the first place. In our proposed SIP model, we have used Functional, Word Embedding, Dense and GRU (Gated recurrent unit), Bi-directional LSTM, Bert to build our model. We have shown that our SIP model is able to determine the suicidal ideation with a higher training accuracy of 88%, a validation accuracy of 89% and training accuracy 98% and validation accuracy 99% from SIP (Sentiment) model.

Index Terms: Suicidal Ideation, Machine Learning, Deep Learning, Neural Network, social media, Functional, Word Embedding, Dense and GRU, Bert, Bi-directional LSTM.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.

Secondly, to our co-advisor Hossain Arif sir for his kind support and advice in our work. He helped us whenever we needed help.

Thirdly, Name and the whole judging panel of Conference Name. Though our paper not accepted there, all the reviews they gave helped us a lot in our later works.

And finally to our parents without their throughout sup-port it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

σ	Sigmoid Function
b	bias of a connection
$BERT$	Bidirectional Encoding Representations from Transformers
CNN	Convolutional Neural Network
f_t	Forget Factor
GRU	Gated Recurrent Unit
i_t	Ignore Factor
$LSTM$	Long Short Term Memory
LTM	Long - Term Memory
RNN	Recurrent Neural Network
SIP	Suicidal Ideation Prediction
STM	Short - Term Memory
$tanh$	Activation Function of tanh
w	weight of a connection

Chapter 1

Introduction

Suicide is an alarming problem all around the world. Every year almost 800000 people commit suicide. Suicide is more common among those suffering from mental illnesses such as sadness and anxiety. Even severe mental disorders without effective treatment can turn to suicidal ideation or suicide attempts. According to studies, suicide is the second-highest cause of mortality among the young population, with a suicide rate of 10.5 per 100000 people. Most of the suicides occur in low-middle income countries, about 79

It is complicated to state the reason why people commit suicide. People with mental health issues or without mental health issues may commit suicide. The tendency of committing suicide depends on many factors such as hopelessness, depressive symptoms, sleep disorders, or violence. Most of the studies in suicide are taking place based on these types of self-report measures of indicator factors or variables. Individual social data could help find suicidal ideation and act as factors of suicidal ideation. Nowadays, people are disclosing their daily life activities on social media, and these data are beneficial to use as a variable to measure someone's mental condition. Previous

studies state that youth of this generation are likely to disclose suicidal thoughts and suicidal factors on social media. Social media made the bigger world into a more miniature world by connecting all the people virtually via the internet. People can now interact with others easily through social media, and this online communication media is the new way of expressing feelings, sufferings, and suicidal tendencies of people. So, these social media are now acting as a surveillance tool for suicidal ideation, and by studying them, we can improve suicide prevention.

Here we can differentiate suicidal thoughts in two different ways, such as suicide planners and suicide attempter. According to some research, most people with suicide planning do not make suicide attempts. Someone with suicidal ideation may express their thoughts of committing suicide in fleeting thoughts, suicide plans, or notes. Some researchers conducted many psychological and clinical kinds of research to detect suicidal ideation. Artificial Intelligence and machine learning can see the prediction of suicide by using the user's data, which can help us understand individual mental states and establish an early intervention. Detection of social content means focusing on content analysis. In order to learn representation, these approaches usually necessitate looking at artificial neural network architectures. The research trend focuses on selecting more valuable features from people's mental health records and developing a neural architecture to better understand the

language with suicidal ideation.
Here we will use some deep learning and neural network methodology to develop an algorithm based on social media like Reddit or Twitter data that can predict suicidal thoughts or ideation. It can serve as a good intervention point. Our main goal is to determine the risk of suicidal intentions or behaviors before tragedy strikes.

1.1 Research Problem

Suicide is becoming more concerning in modern society. It has been an intractable public health problem emails [1]. But it is unclear to find out the comparability with the online expression of suicidal risk as induced by physicians. Still, some research finds a correlation between suicidal ideation expressed online and psycho metrically evaluated suicidal risk [4] [11]. Social sites have become an despite having advanced in diagnosing and treating major mental disorders. Over the past 60 years, the annual suicide rates have been stable at around 10-12 per 100000. A large amount of people does not get any treatment or appropriate care. People suffering from mental health issues sometimes cannot express their problems to others. A study on suicidal found that adolescents are likely to express suicidal ideation verbally and via electronic means, social media sites, blogs, posts, instant messages, and emerging study area in computational linguistic. Thus, we could collect valuable data from these social platforms to develop new technological approaches and improvements that significantly enhance suicide detection and future suicide risk prevention. Furthermore, we can get significant data from users' data to contribute to the analysis of their language preferences. It is important to be determined first to identify the core intention of the user, whether candidates have suicidal ideation or not if we want to work with users' data. Here we need to analyze users' social data using machine learning to predict individuals at risk. Then, to identify or measure existing patterns in users' data, we need to train machine learning techniques to predict suicide risk. As the machine learning approach based on psychological theories of suicide is still undeveloped, we are focusing on using deep learning and neural network approaches such as sequential, word embedding algorithm, Dense and GRU (Gated recurrent unit), Bert, Bi-directional LSTM. It is necessary to analyze the text data first to predict or determine whether a text from a user's social media contains suicidal thoughts or not. We will combine deep learning and artificial neural network approaches to increase the accuracy of determining suicidal ideation via analyzing social media data. We will work with the LSTM and GRU to predict the risk and the ideation of suicide. The prediction can also be determined with fast text word embedding and Bert. Applying these approaches (Bi-directional LSTM, GRU, Bert, word embedding) will bring a difference in the accuracy of predicting the ideation of suicidal thoughts. Therefore, we can come up with a solution and differentiate which approaches would be more consistent to find out the risk of suicide via analyzing social media user data. These approaches would be implemented in the data set of social media text posts. So, we also must keep in mind the execution time of the whole process of predicting suicidal ideation. We would choose that one model or approach that will predict the suicide ideation with the highest possible accuracy and with a lower execution time.

1.2 Research objective

The following research is for detecting depression and suicidal behavior of a particular person to prevent him from doing such action by using deep learning and neural network algorithms. Generally, depressed persons try to hide from surrounding society and usually react with sadness and frustration in their actions. Behavior in their public post on social media can be used to monetize the potential sufferer of immense depression. The objectives of the research are:

1. To thoroughly understand the human behavior in depression and how they react.
2. To thoroughly understand the algorithms of deep learning and neural network.
3. Another entry in the list
4. To invent and develop a model SIP to detect Suicidal behavior using BERT embedding, BERT preprocessing, functional model and deploy it.
5. To express the model and recommendations on the model.

Chapter 2

Literature Review

To detect the intention of suicide, many researchers have come with different approaches in recent years. These approaches have been proposed with the objective of automated detection of suicide ideation, ranging from traditional methods like logistic regression [9] [2] [10], SVMs [2] to deep learning classifiers like CNNs [12] [6], LSTMsCNNs [21] [17] and their combination. Many publications that have worked with the information from social media have used Twitter data, but it was pretty limited. Besides, most research evaluated the Twitter data, which only mentioned suicidal thoughts or attempts. There was a lack of factors or indicators of the tweet data set that could predict the risk of suicidal thoughts or ideation. In a recent standard review of publications in applying technologies for suicide prevention, Martin et al. found only 30 papers, of which 12.9 percentage were related to social networks, and 3.23 percentage were related to machine learning [13]. Despite the advanced technological solution to identify, the authors stated that there are yet such innovative specialized tools and predict the suicide risk. Braithwaite et al. evaluated Twitter profiles of 135 study participants validated a machine learning approach using the Linguistic Inquiry and Word Count in short LIWS process to predict suicidal individuals with 92 percentage accuracy [5]. In 2017, O'Dea et al. used Linguistic Inquiry and Word Count analysis in Twitter data to demonstrate that strongly concerning suicide-related posts had distinctive linguistic profiles [8]. In 2018, Du et al. applied ML to generate a convolution neural network to identify suicidal tweets [3]. Unfortunately, this method worked only when suicide is mentioned in the tweet and creates an opportunity to intervene in those at risk of suicide attempts. However, these approaches are not enough and cannot predict suicidal thoughts or ideation before development, as a suicidal tweet must be written for the model to identify it. Thus, these approaches may identify suicidal in cases that do not tweet about suicidal thoughts. Burnup et al. have developed a machine learning classifier to identify individuals with suicidal ideation on Twitter with 68-73 percentage accuracy [7]. Besides, it is not clear whether the technology contains individual suicidal thoughts to generate predictive accuracy, which creates a massive lack in the field and an excellent opportunity to use AI in suicide prediction.

In the field of suicidal intention identification, in this paper [18] S. Fodeh et al., worked with Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA), Non-Negative Matrix Factorization (NMF). Besides, the authors of this paper use topic identification to identify the topics on the tweeter data whether it has some data for their model by LSA, LDA, NMF. They assume 12 suicide-related topics to identify risk factors according to predefined 12 risk factors. The user data then was identified from the output of LSA, LDA, NMF according to specified 12 risk factors. Furthermore, to classify users into groups such as “AtRisk” and “HighRisk” they use K-means clustering. The author uses K-Mean and Decision Tree to distinguish the users between “AtRisk” and “HighRisk”. Their motive is to classify the user into “AtRisk” and “HighRisk” using K-Mean clustering and decision tree based on the domination of the risk factor. The model will labialize the user as “AtRisk” if it identifies the cluster has mostly that specific user and the same will be happened for “HighRisk”. Here, after clustering and performing decision tree calculation they found none of these can clearly distinguish the user as “AtRisk” and “HighRisk”. Decision Tree was likely to classify the user as “HighRisk” and here the sensitivity was close to one and the specificity was close to 1. On the other hand, K-Mean clustering was likely to classify the user as “AtRisk” and the sensitivity was close to zero and the specificity was one. Therefore, the authors found a way in which they use Decision Tree using the K-Mean approach in a balanced dataset and they found the best result by applying this approach. They found 58.6 percentage accuracy using the NMF algorithm with the K-Mean approach whereas the accuracy of Decision Tree using the K-Mean approach is 88.5 percentage in the NMF algorithm. Tadesse et al. in this research paper [14], they used LSTM and CNN combined as LSTM-CNN to detect suicidal ideation detection in social platform Reddit. They used the output result of LSTM as input of CNN thus they built a new type of CNN on the base of LSTM to improve the classification result. They classified the suicidal and non-suicidal content by the hybrid framework of LSTM-CNN. There were some layers such as Word Embedding Layer, Dropout Layer, Long Short-Term Memory (LSTM) Layer, Convolution Layer, Pooling Layer, Flatten Layer, and finally Output Layer. Word Embedding Layer is responsible for assigning a unique index to a word of sentence for forming a fixed-length vector. In this layer, they used the category of shallow model’s Word2vec. In this, they trained 2 layers to reconstruct word context from the surrounding word window. To avoid over-fitting, they used a Drop-out layer. In this layer, they randomly drop out the noise and prevent the co-adaption of hidden units. To find long-distance dependency in the text they used the LSTM layer they used 100 LSTM units in a single layer. They used convolution layer for feature extraction. Here, to create feature dimension pooling layer group the information and later this was converted to column vector by flattening layer. In the output layer, the final result with help of the neural network process was obtained by using the SoftMax function. They use “ReLU” as their activation function in convolution layer and pooling size, units, embedding dimension, batch size, number of epochs, dropout, the fully connected layer was max-pooling, 100, 300, 8, 10, 0.5, SoftMax respectively. In the LSTM and CNN method with the feature type of Word2vec, they got an accuracy of 91.7 percentage and 90.6 percentage with a precision of 94.8 and 91.8 respectively. But they got much more accuracy with the combined modified framework of LSTM-CNN which is 93.8 percentage with a precision of 93.2. Roy et al. in this paper [19], their main motive was to develop a model to detect an individual’s risk of suicide and also detect the development of suicidal ideation into an individual within a time frame. They constructed their ML model upon interpersonal psychological theory for suicide (IPTS), the hopelessness model, depression, anxiety vector, and insomnia with suicidal ideation. They developed their ML model called “Suicide Artificial Intelligence Prediction Heuristic

(SAIPH)" to identify suicidal ideation in tweeter data. Firstly, they identify Suicidal ideation (SI) tweets in the identification of SI tweets. Here they scanned the user's timeline to identify the presence of any suicidal indicative word. They use predefined word patterns to identify whether the tweet was suicidal or not. If the tweet contained any of the word patterns, then the model classifies the tweet as an SI event. Then they evaluated this data by psychologists to be confirmed that these data represent Suicidal ideation. The algorithm used a two-tiered approach to accomplish two goals. The first goal was to develop a system for distinguishing SI cases from controls, and the second goal was to develop a mechanism for determining whether highlighted persons are in danger. The algorithm formation for objective one entails first using neural networks to convert word-based tweet data into scores representative of psycho-social variables and then training a random forest on the resultant of neural network data in a training data set controls to predict case status. In order to achieve objective two, tweet-level random forest prediction scores created across a user's timeline are analyzed for changes in the frequency of high scores in comparison to the user's typical pattern. They got an accuracy of 88 percentage when the user is female and predict N=300 Suicide ideation from N=937. Besides, the model predicted N
= 37 SI events with an accuracy of 0.75 where the actual SI event of individual suicide plan (SAP) N = 97. Shini Renjith et al., in this paper [20] used the similar approach as the above-mentioned paper [19] but in a different way. They categorized the label into four i.e. No Risk, Low Risk, Moderate Risk, Severe Risk. They use a data set of 11000 people who post in subreddit at least once. In their proposed model they used LSTM and CNN combined as this paper [18] but they used attention model alongside this LSTM-CNN model and built LSTM-Attention-CNN model. Unlike paper [18] this model used the result vector of LSTM for the input of Attention Layer and the output of Attention layer was the input of CNN layer. They use Natural Language Toolkit (NLTK) for data preprocessing. They use tokenization to tokenize the concatenated user post. Then they applied normalization on it. Besides they used lemmatization to reduce senseless word item in the dataset. Then after all the preprocessing the clean data was fed to word embedding layer. The procedure after this as same as this paper [18] with the difference in one layer. They introduce attention layer after LSTM layer where they used tan hyperbolic function and normalized with SoftMax function. This SoftMax function was introduced as activation function here. In this model they used a 300-dimensional pre-trained Word2vec. The data set carried 69600 post which was splint into train and test by the ratio of 0.8 and 0.2. Adam optimizer was used for bias-correction operation for optimization when the gradient became sparse, and it also helped them to converge faster. The proposed model got 90.3 percentage accuracy with a precision of 91.6 percentage whereas LSTM-CNN, LSTM, CNN model got 88.2 percentage, 87.7 percentage, 86.6 percentage with a precision of 88.3 percentage, 90.8 percentage, 87.8 percentage respectively. Here the proposed model got better accuracy with better precision.

S. Jain et al., in this paper [16] took 2 datasets as their input to the model, one of the data sets is from answering question and the other from the tweeter reddit data. 1st data set was built from student and the other was built from the user post on twitter. For data set one they defined an index value for every answer from which the final processing would be calculated. The choice of the answer was encoded by label encoder where they categorized the answer in 4 with the value of 0 to 3. They took positive and negative sentimental labeled data set from Kaggel. Then they remove non-alphabetic characters, stopwords by using NLTK stopwords corpus. For feature extraction for data set-1 they split the data in the ratio of 0.8 and 0.2 where 80 percentage of the data is for training purpose and the rest was used for test purpose. For data set-2 they use Term Frequency Inverse Document Frequency (Tf-idf), weighted word count for feature extraction. Like data set-1 they split the data in ratio of 0.8 and 0.2 where 80 percentage data was reserved for train purpose and 20 percentage was used for test purpose. They categorized the depression label into 5 i.e., minimal, mild, moderate, moderately severe, severe with the sore of 0-5, 5-14, 14-27, 27-39, greater than 30 and the stage was 0 to 4 respectively. To detect suicidal ideation they use logistic regression, Random Stage Classifier, XGBoost Classifier, SVM (Support Vector Machine). This classifier was responsible for detect any symptom of suicidal ideation in the user data. For data set-1 they got the highest accuracy in XGBoost Classifier which is 83.87 percentage and for data set-2 they got the highest accuracy in Logistic Regression Classifier which is 86.45 percentage.

Chapter 3

Methodology

In our proposed Suicidal Ideation Prediction (SIP) model, we tried to detect suicidal intent from textual data which are publicly available in twitter and reddit. This model will help to detect suicidal ideation and using the model in social media apps will help preventing suicide. For the detection process, our model requires a designing process that takes input from data sets (publicly available sentiment data sets, most of them are collected from Kaggle) and after some systematic process it generates an output(predicted).

3.1 Datasets

The data sets we used in our SIP are Kaggle data sets (Twitter, Reddit, IMDB sentiment data). The data sets contain textual data (tweets, status, opinions). There are three classes in the data set. The columns of the data set are described below.
Dataset 01:

Column Name	Description
Sentence	Text data
Sentiment	Classes (negative, positive, neutral)
Sentiment Encoding	Encoded classes (0, 1, 2)
Suicidal	Classes (suicidal, non-suicidal)
en-suicidal	Encoded classes (0,1)

Table 3.1: Dataset description table 01

Dataset 02:

Column Name	Description
text	Text data
class	Classes (negative, positive, neutral)
Encoded class	Encoded classes (0, 1)
Suicidal	Classes (suicidal, non-suicidal)

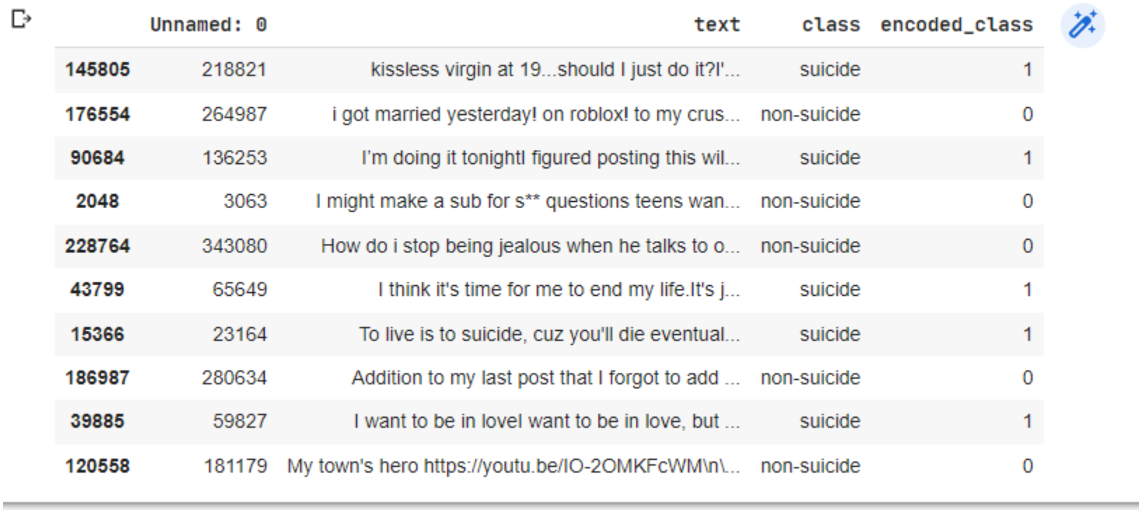
Table 3.2: Dataset description table 02

We have worked with two data sets, and they are given bellow with example Dataset 01:

	Sentence	Sentiment	Sentiment Encoding	Suicidal	en_sucide
87252	AITA for trying to kill myself?\n\n\nI recentl...	Positive	1	Yes	1
87253	Im trying to find a reason to stay, but its ha...	Positive	1	Yes	1
87254	I'm worthless. Shouldn't exist. No one will mi...	Positive	1	Yes	1
87255	I needed something uplifting after this horrif...	Positive	1	No	0
87256	Below are the top 3 underappreciated stories o...	Positive	1	No	0
87257	In the past few days lots of non-WoahDude mate...	Positive	1	No	0
87258	And in 2014, June 13th is going to land on a F...	Netural	0	No	0
87259	So our genes are passed on from generation to ...	Positive	1	No	0
87260	If you want a reminder on the day of the premi...	Positive	1	No	0
87261	|APP NAME|iOS|Android|other OS|PRICE|WoahDude ...	Positive	1	No	0
87262	I really appreciate the time that someone took...	Negative	2	No	0
87263	I wanted to share my ayahuasca experience that...	Negative	2	No	0
87264	https://youtu.be/S1KfuErAcj0	Netural	0	No	0
87265	You can have fictional people, fictional food,...	Netural	0	No	0
87266	#IMPORTANT\n\n#Please read even if you're not ...	Positive	1	No	0

Figure 3.1: Dataset description tables

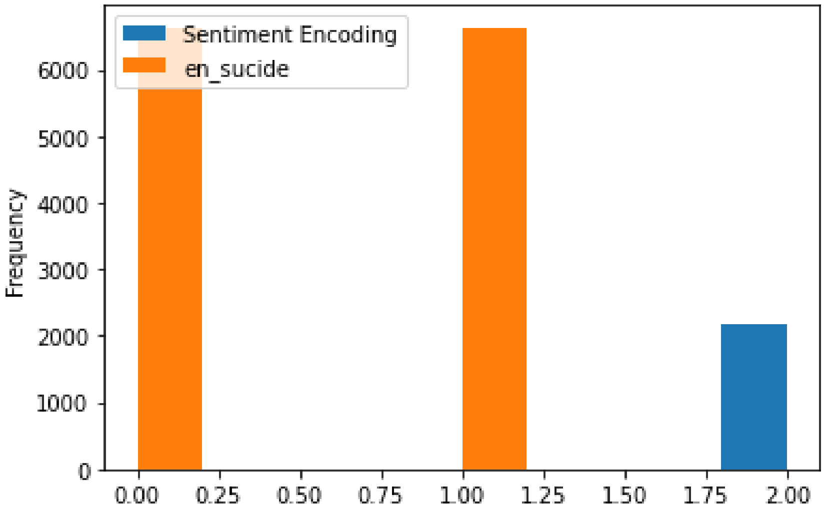
Dataset 02:



	Unnamed: 0	text	class	encoded_class
145805	218821	kissless virgin at 19...should I just do it?I'...	suicide	1
176554	264987	i got married yesterday! on roblox! to my crus...	non-suicide	0
90684	136253	I'm doing it tonightI figured posting this wil...	suicide	1
2048	3063	I might make a sub for s** questions teens wan...	non-suicide	0
228764	343080	How do i stop being jealous when he talks to o...	non-suicide	0
43799	65649	I think it's time for me to end my life.It's j...	suicide	1
15366	23164	To live is to suicide, cuz you'll die eventual...	suicide	1
186987	280634	Addition to my last post that I forgot to add ...	non-suicide	0
39885	59827	I want to be in loveI want to be in love, but ...	suicide	1
120558	181179	My town's hero https://youtu.be/IO-2OMKFcWM\n\...	non-suicide	0

Figure 3.2: Snapshot of the data sets after preprocessing and label encoding

The distribution of the features in the data set
Dataset 01:



Dataset 02:

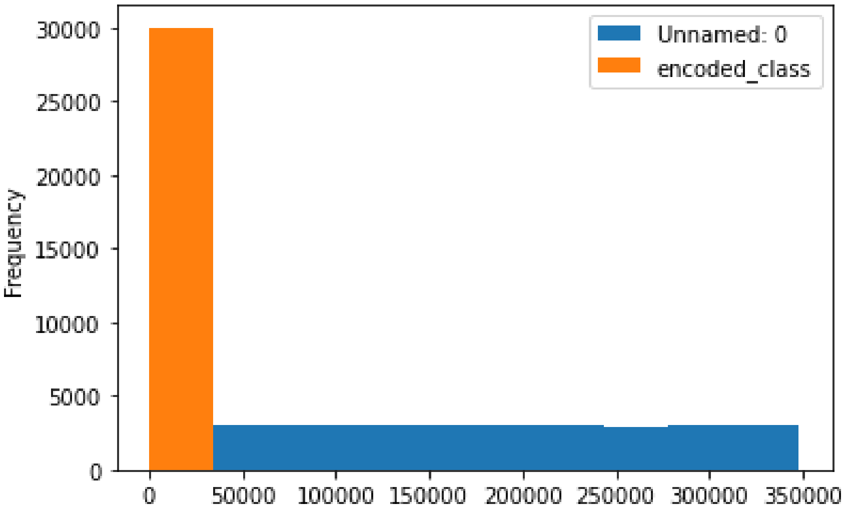


Figure 3.3: Distribution of the data in data set

3.2 Proposed Model (SIP)

SIP (Suicidal Ideation Prevention) is the model where we are taking publicly available Twitter/Reddit data in our case from Kaggle in CSV format as input. In the next step we try to understand the properties of our data sets, so we use data visualization, then we only select the trainable data. In the next step, we preprocessed and balanced the data set. In the next step, we normalize (text normalization) and encode the data set using the dataset process. Then we use BERT word embedding to get vectors (The numerical form of the textual data saved the word embedding for finding the cosine similarity. Then, we split the data set 80% and 20% for training and validation purposes. Then we train our data sets and evaluate the data sets and make decisions from input text on whether the individual has any suicidal ideation or not. If the output is 1 then the individual has suicidal ideation, if 0 then the individual has no suicidal ideation.

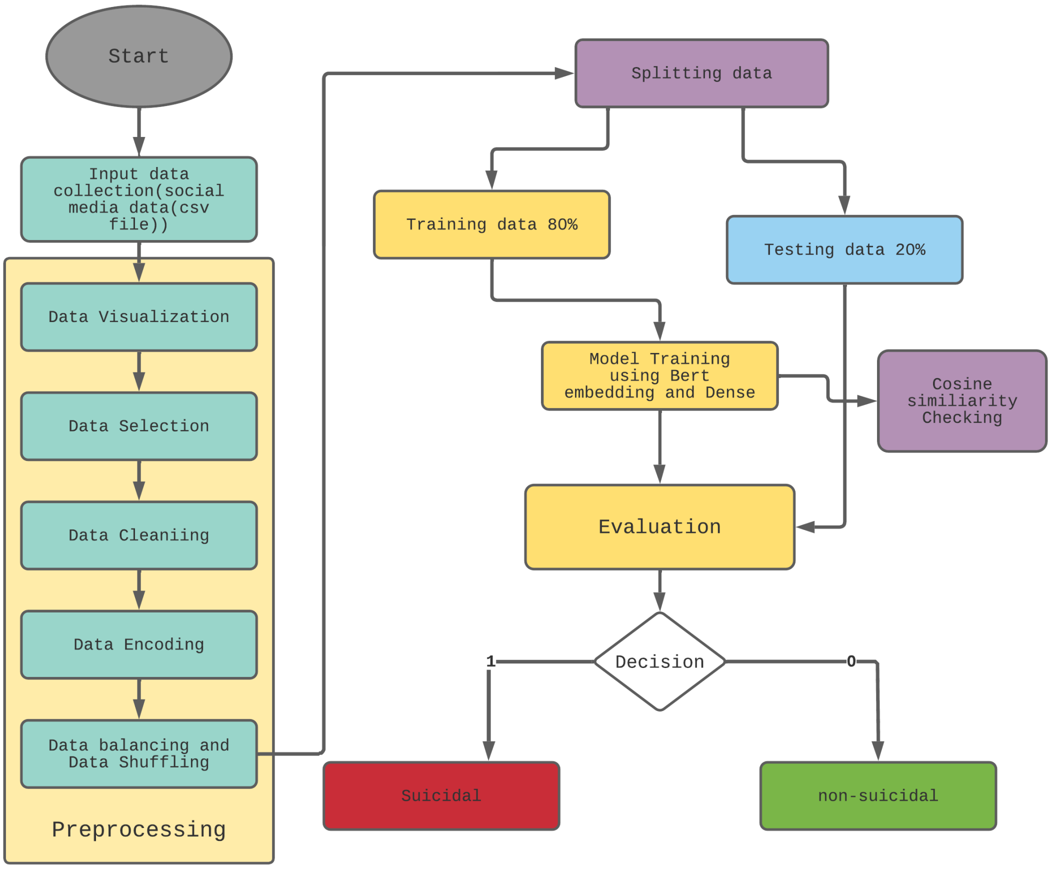


Figure 3.4: SIP model

3.3 Data Preprocessing

The datasets from Kaggle that we used were raw. We transform our dataset into an appropriate form so we can build a very efficient model for our detection. This dataset contains more than 30000 text data with sentiment levels. The text data contains websites, hashtags, mentions, emails, and emojis which could create problems while training. In our dataset, we also had some meaningless words and punctuation marks which we had to remove. We followed some of the steps for preprocessing and data-cleaning for making our dataset usable for our model and we had to encode the labels too.

Preprocessing steps:

3.3.1 Data Reduction

We removed the rows with NaN values in the sentiment level column and we also removed the text data with blank row after cleaning. Data Reduction helped us to reduce the number of unnecessary rows from the data set.

3.3.2 Data cleaning

As we are working with text data from twitter and reddit. The texts were mixed with emails, URLs, hashtags, and mentions. So, we follow some steps for cleaning the data. At first, we lower case all the data for making them easier for future text normalization. Then we remove the noises (emails, URLs, mentions,) from the data using regular expression. Then we tokenize the data, so sentences convert into words. Then on the next step we remove stop words (English) and punctuation.

3.3.3 Data Normalization (Text Normalization)

For making our model easy to train this data normalization step was very important. We use BERT preprocessing and encoding for normalizing the text data. We used stemming and lemmatization for data normalization when we train Bi-LSTM and Bi-GRU.

3.3.4 Label Encoding

We use label encoding for the label column that took all the classes (category) in numerical values.

Chapter 4

Model Architecture

4.1 GRU

GRU (Gated Recurrent Unit) is a form of neural networks that aims to solve vanishing gradient problem and exploding gradient problem of RNN (Recurrent Neural Network) and slow and expensive memory operation by cell-state of LSTM (Long Short-Term Memory). Unlike LSTM GRU do not need three gates. GRU only requires two gates. They are Update gate and Reset gate

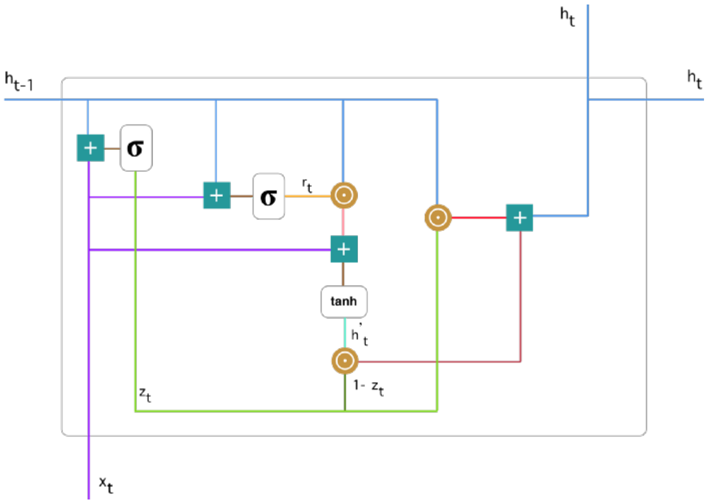


Figure 4.1: GRU

Update Gate: Just like the forget gate in LSTM the Update gate in GRU helps the model in our case SIP model how much information to keep from the previous step. It uses a sigmoid function of range [0,1] so, it can pass 0 percentage to 100 percentage past information. The equation of update gate zt ,

$$Z_t = \sigma\{(W_z \times X_t + U_z \times h_{t-1}) + b_r\} \tag{4.1}$$

Reset Gate: Reset Gate is almost same as the update gate, but the difference is the update gate decides how much data to keep from the previous step, but the reset gate decides how much past information to forget or reset. The equation of reset gate,

$$r_t = \sigma\{(W_r \times X_t + U_r \times h_{t-1}) + b_r\} \tag{4.2}$$

Current memory content, ht

$$\varphi_h(x) = tanh(x) \tag{4.3}$$

$$\hat{h}_t = \varphi\{W_h \times X_t + U_h \times (r_t \odot h_{t-1}) + b_h)\} \tag{4.4}$$

$$h_t = \varphi\{(1 - Z_t) \odot (h_{t-1})\} + (b_h \odot \hat{h}_t) \tag{4.5}$$

$$\hat{y}(t) = \varphi(W_h \times \hat{h}_t) \tag{4.6}$$

4.2 Word Embedding

This is a technique of natural language processing which is used to analyze text. Each word in a text data represents by real number. The set of the real numbers which contains unique word is a vector and n dimensional vectors would be helpful to find out the word meaning in their values means that we encode the meaning of a word from the text data. A word embedding is a learnt text representation in which words with related meanings are represented similarly. The technique word embedding follows is that it represents each word as real valued vectors. This representation of each word as real valued vectors occurs in a predefined vector space. By using this technique, we can improve our accuracy while analyzing each word from the text data to find out whether the text contains suicidal ideation or not. It is very much important to determine the actual meaning of text data as based on that our proposed model would determine the ideation of suicide. Word embedding comes up with the same representation for the similar type of words means if some words have the same meaning, then their word embedding represent those words in the same way. As a result, we can capture the same meaning of different words. We have used embedding layer on the front end of our neural network model. There are two popular word embedding methods. Which are Word2Vec and GloVe.

4.3 BERT

Bert stands for Bidirectional Encoding Representations from Transformers. The transformer was first introduced by google to replace Recurrent Neural Network (RNN) and Convolutional Neural network (CNN) in the field of Natural language modeling in 2017. Even though texts are ambiguous, BERT is used for helping the computer understand the meaning of natural language in the form of plain text. To understand the ambiguous text BERT use surround text. BERT uses a deep learning model called transformer in which the input and output are connected, which means output from a node is the input element for every input node in the next run. The weight between them is also calculated dynamically by the process called "Attention" based on the connection of input and output elements. Thus, it can work bidirectionally, which means BERT can read the text in left to right and right to left both ways at the same time whereas the other model can only read in one direction from left to right or right to left sequentially. To perform this BERT is therefore pre-trained on two NLP Tasks called: Masked Language Modeling (MLM) and Next Sequence Prediction (NSP) by using the bidirectional capabilities. The masked Language Model (MLM) is used for hiding a word in the sentence in order to predict it by the program by using the context of the hidden word. Next Sequence Prediction is used for the prediction of the connection between two sentences whether it is logical, sequential or random.

BERT was trained with a form of unlabeled plain text. Even when it is implemented in actual applications, it continues to learn unsupervised from unlabeled text and improve. The pre-training data helps to build the knowledge. BERT may then be fine-tuned to a user's demands and adapt to the ever-growing collection of searchable material and queries. This is called transfer learning.

BERT's greater ability for grasping meaning and ambivalence in language is due to the transformer. Instead of processing each word individually, the transformer processes each word in context to all other words in the statement. By looking at all neighboring words the Transformer helps the BERT model to identify the whole context of a word. This allows the BERT for better understanding the language context.

BERT is the first-ever NLP approach that depends purely on the self-attention mechanism, which is enabled by the bidirectional Transformers at its core. This is necessary because a word's meaning can frequently change as the conversation progresses further. Each new word adds to the overall meaning of the term that the NLP algorithm is focusing on. The more words in a statement or phrase, the vaguer the targeted word becomes. BERT provides the mostly accurate meaning of a word by considering the influence of all other words in a sentence on the targeted word and minimizing the left-to-right influence that can change the meaning of the word as a sentence proceeds and this was possible for bi-directional capability of BERT.

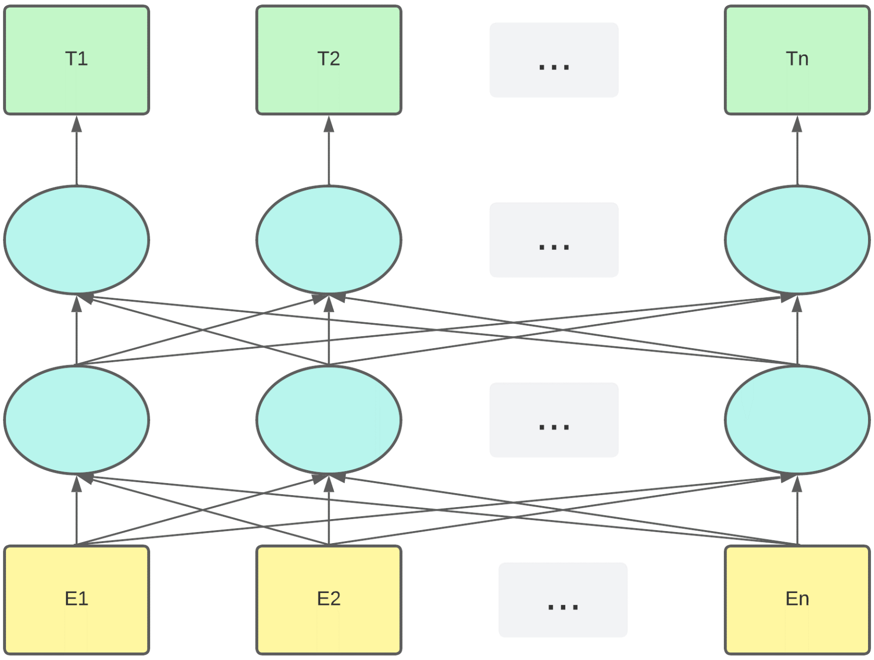


Figure 4.2: Figure of BERT

4.4 LSTM

LSTM stands for long-short term memory which is one type of Recurrent Neural Network (RNN). Traditional feed forward Neural Network like Convolutional Neural Network (CNN), GANs can't memorize the past action. Means CNN, GANs have no memory. But RNN or specifically LSTM can recall past data or training information. Generally, in CNN or GANs if we feed the neural network model with the input and get the output, the output will never be used in the future. Means the neural network will forget the present output for a particular input. Therefore, the output cannot be repurposed for the improvement of the model. But in RNN the output of a specific input can be used as an input in the future to improve the outcome of the model.

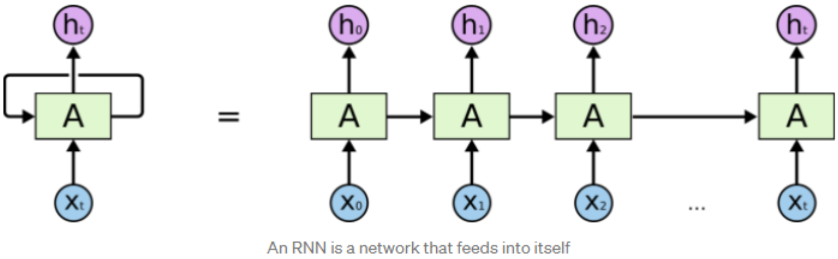


Figure 4.3: Figure of RNN
[15]

We normally use RNN when we need the model to remember past output to understand each word of a sentence. In RNN each word of a sentence is considered as separate input and the network will keep the most weight int the latest input. But this creates a significant problem called "Vanishing Gradient". Means, the network will forget the first input in the process of relevance selection because the weight of the first input will be zero this time. This problem can be addressed by introducing of LSTM (Long-Short Term Memory).

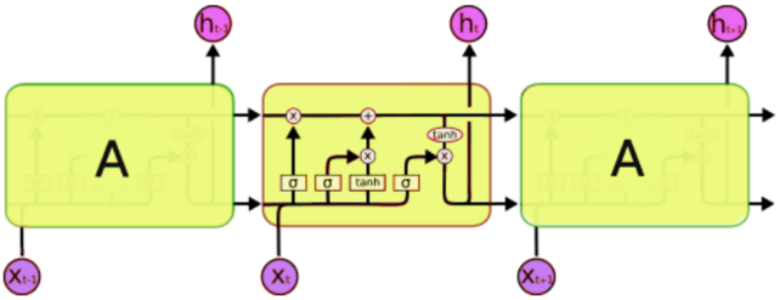


Figure 4.4: Figure of LSTM
[15]

LSTM is basically a special type of RNN which does a lot of Mathematical calculations to have a better memory then normal RNN. There is no such calculation in traditional RNN, instead it feeds the outcome of an input in the next part of the network. But in LSTM there are four gates to have a better memory:

1. Forget gate.
2. Learn gate.
3. Remember gate

Besides, there is three inputs as well. Those are:

1. Long term Memory (LTM) info.
2. Short term memory (STM) info.
3. Some training data "E"

When these three data are fed into LSTM they go through either "Learn gate" or "Forget gate". The long-term memory info will be gone through the forget gate where some parts of the sentence will be forgotten which are irrelevant. Though this forgotten part will be in forget gate instead of actually forgetting it. The short-term memory info and the training data "E" will be gone through the "Learn gate". The purpose of this gate is to decide which part of the sentence will be learned. Then the value of forget gate and learn gate go through the "Remember gate". This gate is responsible for making the new long – term memory. The output of the network comes from the "Use gate" which is also responsible for the update of the short – term memory. Brief discussion on the gates is mentioned bellow:

Learn Gate:
This gate gives the ignore factor (i_t) and (N_t).
The equation of (N_t) is:

$$N_t = tanh(W_n[STM_{t-1}, E_t] + b_n) \tag{4.7}$$

Here in this gate, combination of the present short – term info with some training Data (E) multiplied by a matric (w) and then adds the bias (b) will be calculated. Then finally the value will be fed into tanh activation function.

The equation of ignore function:

$$i_t = \sigma(W_i[STM_{t-1}, E_t] + b_i) \tag{4.8}$$

To calculate the ignore factor, it needs to combine the existing short - term memory data with the training data and multiplied be a new weight (W_i) and add them with a new bias (b_i). This calculated value then feed into a sigmoid function. When the ignore factor (i_t) and N is calculated the output of the learn gate will be:

$$Output(learngate) = ignorefactor * N$$

This output is the new learned information for training data (E) for the next part of the neural network.

Forget Gate:
Forget gate is responsible for forgetting the long – term unnecessary information of the sentence. There is a forget factor (f_t) which help to forget some irrelevant long – term info. This forget factor will be multiplied with the 'LTM' (Long – term memory) in order to forget the irrelevant long-term memory.
The equation of the forget factor is:

$$f_t = \sigma(W_f[STM_{t-1}, E_t] + b_f) \tag{4.9}$$

Remember Gate:
This gate is responsible for computing new long – term memory. To do this the value of the forget gate and the value of the learn gate will be combined together. The output of remember gate is: Output of learn gate + output of forget gate

Use gate:
Use gate is responsible for providing the output or the new short – term memory. It gives the (U_t) and (V_t) from the value of forget gate and learn gate to calculate the output or short – term memory.

The equation of (U_t):

$$U_t = tanh(W_u LTM_{t-1} f_t + b_u) \tag{4.10}$$

The output of forget gate adds with the bias (b_u) and then goes through the activation function tanh.

The equation of (V_t):

$$V_t = \sigma(W_v[STM_{t-1}, E_t] + b_v) \tag{4.11}$$

The output of learn gate combined with the new bias (b_V) will be gone through the sigmoid function to calculate (V_t).
To get the new short – term memory final the (V_t) and (U_t) will be multiplied in this gate.

New short – term memory = (V_t) * (U_t)

Here in our thesis, we use bidirectional LSTM instead of normal LSTM. In bidirectional LSTM instead of one model two model are used for better memory. One model is used for learning the provided input sequence and the other model is used for learning the reverse of that sequence. In bidirectional LSTM there is a term called “Merge Step”. Merge step is used for merging the two model in bidirectional LSTM. The merging can be performed by summation, multiplication, averaging, concatenation etc.

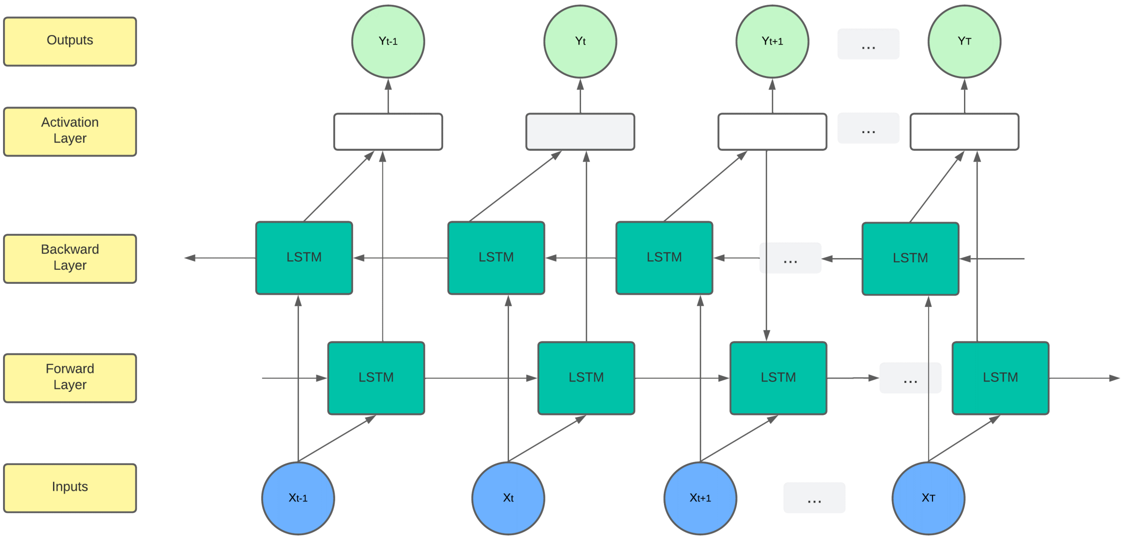
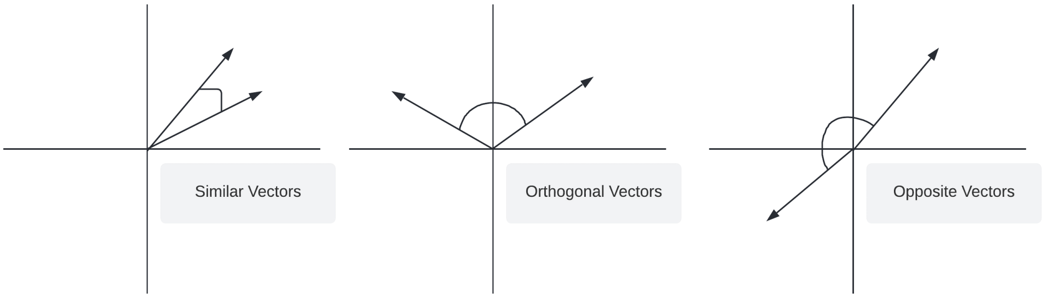


Figure 4.5: Figure of bi-directional LSTM

4.5 Cosine similarity

Cosine similarity is a measurement or metric used to measure the cosine or the similarity of the angle between two or more vectors. The cosine of the angle between vectors is the cosine similarity. The vectors are usually non-zero and belong to an inner product space. Belong to the inner product space means that they must create a scalar through the inner product multiplication. This measure of similarity is more concerned with direction that with magnitude. In other words, two cosine vectors aligned in the same direction have similarity of 1, whereas two vectors aligned perpendicularly have a similarity of 0. Then the similarity measurement would be -1, when two vectors are diametrically opposed. It means that they are positioned in exactly opposite directions. Furthermore, cosine similarity is frequently used in positive space between 0 and 1. Cosine similarity is primarily concerned with similarities in orientation and does not consider or assess differences in magnitude.



The cosine similarity is defined mathematically as the dot product of the vectors divided by magnitude of each vector. For instance, if there are two vectors, A and B, then the similarity between them would be calculated as:

$$Similarity = cos(\theta) = \frac{A.B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4.12}$$

Cosine similarity is used in different sectors of machine learning. The application of the cosine similarity could be in data mining, information retrieval, word documentation in natural language processing, transaction in market data and recommendation system, images in computer vision and so on. It can also be used as a loss function at the time of training neural networks.

4.6 The functional model:

The functional model is a method of building deep learning in our case NLP model. This model has some benefits over the Sequential model. Unlike the sequential model the functional model helps us to reuse the separately created layers and sharing the inputs and the outputs from the layer. In SIP model the functional model help getting the vectors from the embedding layer that could be used for finding the cosine similarity.

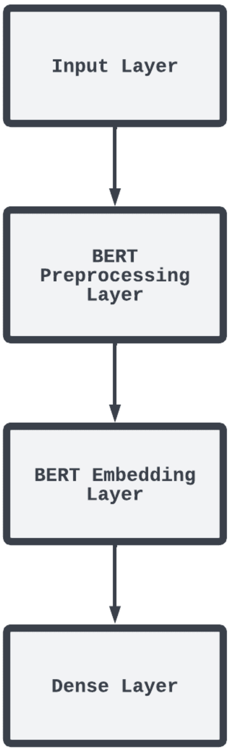


Figure 4.6: functional model

4.7 Model Summary of SIP

Our proposed model SIP is a functional model. We used SIP. We trained our proposed SIP(Sentiment) and SIP(Suicide) model with 10 epochs. We added one embedding layer(BERT), two Dense layers. Batch size was 16 and we used 'adam' optimizer. The at the output neuron we used "Sigmoid" activation function in case of SIP(Suicide) and on the output neuron we used "softmax"

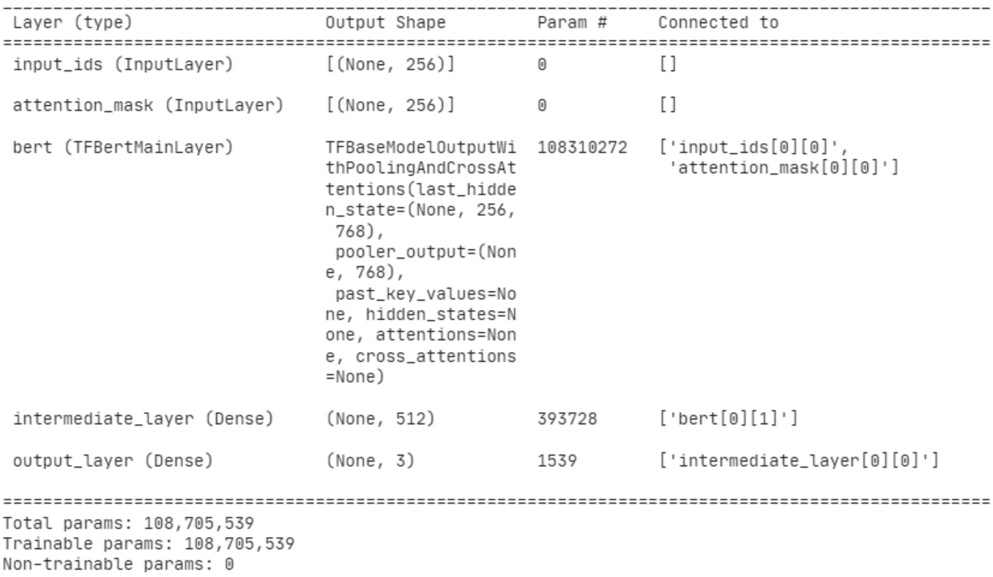
Activation function: Activation function is a function that determines which function will fire and which function will not it uses weighted sum method for calculating it. After calculating the weighted sum, it compares the value with the threshold value. Our problem is a multi class classification problem, so we used 'softmax' as activation function.

Loss function: The loss function helps us to calculate the difference between the true and predicted value. For binary classification there are a lot of loss function available. We used categorical cross entropy and sparse categorical cross entropy while testing sparse categorical cross entropy gave us more accuracy. So, we used sparse categorical cross entropy in SIP.

Accuracy: The number determines how successful the model in case of predicting. We calculate accuracy by dividing the Number of correct predictions by total number of predictions. We evaluate different kind of accuracy such as training accuracy, validation accuracy, test accuracy.

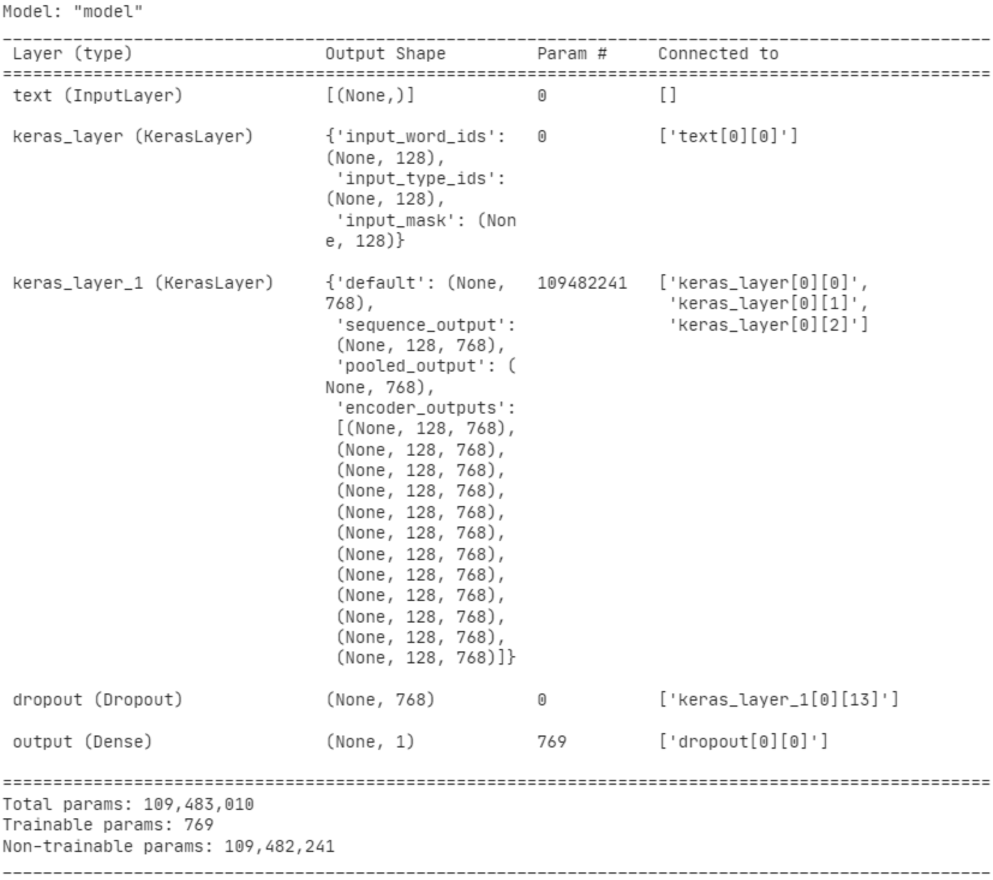
Optimizer: The optimizer helps the model updating the learning rate and weight. For every iteration optimizer is the most important function for helping the model to train properly. We used 'adam' optimizer in our proposed model.
The model summary:

We used 30000+ text data as our input we in our input layer, then we use BERT for preprocessing and embedding. Then we use dropout layer for preventing over fitting. Then we used Dense layer for training. In our case, we saw that Bi-LSTM and Bi-GRU is prone to over fitting even after regularization. So, we used BERT here.



```
____________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
====================================================================================================
 input_ids (InputLayer)         [(None, 256)]        0           []

 attention_mask (InputLayer)    [(None, 256)]        0           []

 bert (TFBertMainLayer)         TFBaseModelOutputWi  108310272   ['input_ids[0][0]',
                                thPoolingAndCrossAt               'attention_mask[0][0]']
                                tentions(last_hidde
                                n_state=(None, 256,
                                 768),
                                 pooler_output=(Non
                                e, 768),
                                 past_key_values=No
                                ne, hidden_states=N
                                one, attentions=Non
                                e, cross_attentions
                                =None)

 intermediate_layer (Dense)     (None, 512)          393728      ['bert[0][1]']

 output_layer (Dense)           (None, 3)            1539        ['intermediate_layer[0][0]']

====================================================================================================
Total params: 108,705,539
Trainable params: 108,705,539
Non-trainable params: 0
```

Figure 4.7: Model (SIP(Sentiment))



```
Model: "model"
____________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
====================================================================================================
 text (InputLayer)              [(None,)]            0           []

 keras_layer (KerasLayer)       {'input_word_ids':   0           ['text[0][0]']
                                (None, 128),
                                 'input_type_ids':
                                (None, 128),
                                 'input_mask': (Non
                                e, 128)}

 keras_layer_1 (KerasLayer)     {'default': (None,   109482241   ['keras_layer[0][0]',
                                768),                             'keras_layer[0][1]',
                                 'sequence_output':               'keras_layer[0][2]']
                                 (None, 128, 768),
                                 'pooled_output': (
                                None, 768),
                                 'encoder_outputs':
                                 [(None, 128, 768),
                                 (None, 128, 768),
                                 (None, 128, 768),
                                 (None, 128, 768),
                                 (None, 128, 768),
                                 (None, 128, 768),
                                 (None, 128, 768),
                                 (None, 128, 768),
                                 (None, 128, 768),
                                 (None, 128, 768),
                                 (None, 128, 768),
                                 (None, 128, 768)]}

 dropout (Dropout)              (None, 768)          0           ['keras_layer_1[0][13]']

 output (Dense)                 (None, 1)            769         ['dropout[0][0]']

====================================================================================================
Total params: 109,483,010
Trainable params: 769
Non-trainable params: 109,482,241
____________________________________________________________________________________________________
```

Figure 4.8: Model (SIP(Suicide))

Chapter 5

Implementation and Results

On the previous chapters we discussed the process we used for collecting, cleaning, balancing basically the preprocessing steps. Now in this chapter we will discuss about the SIP model and its implementation with brief explanation. Figure [3.4] contains the description of our SIP model.

5.1 Implementation:

We use python as our primary programming language. For array and matrix manipulations we used NumPy. We used Pandas for csv file operations and data analysis. We used BERT preprocessing for preprocessing the data and we used BERT word embedding for word embedding. Regular expressions were used for cleaning the data. Plots were generated using the matplotlib and seaborn. We used plots for data visualization. We used TensorFlow and tensor board for training and accuracy showing purpose. We used Bert and Dense for building the SIP model from the scratch.

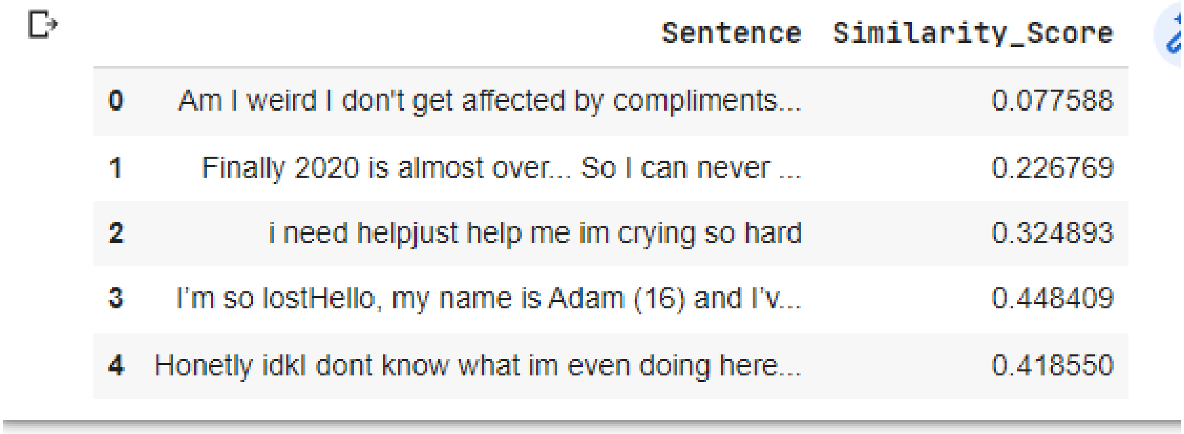
Bert-Embedding Layer: Bert embedding layer is responsible for forming vectors. The vectors are actually unique number for the words with similar meaning. This layer converts word to their numerical form and later tokens where a word starts with CLS and ends with SEP. This word embedding can be used for finding the cosine similarity as well.

1. **Input Layer:** In our SIP model we used the Input layer in karas. The input layer takes shape of the input, data type of the input and the name.
2. **Input Layer:** In our SIP model we used the Input layer in karas. The input layer takes shape of the input, data type of the input and the name.
3. **Pre-process Layer:** On the pre-process label we used Bert-preprocess for preprocessing the input texts. This layer creates attention masks from the input ids.
4. **Dropout Layer:** The dropout layer helps to prevent over fitting at the time of training.

5. **Dense Layer:** This layer works as a fully connected neural network. In our SIP(Sentiment) we used two Dense layer and SIP(Suicide) model we used a Dense layer. The last Dense layer in our SIP model works as the output layer.

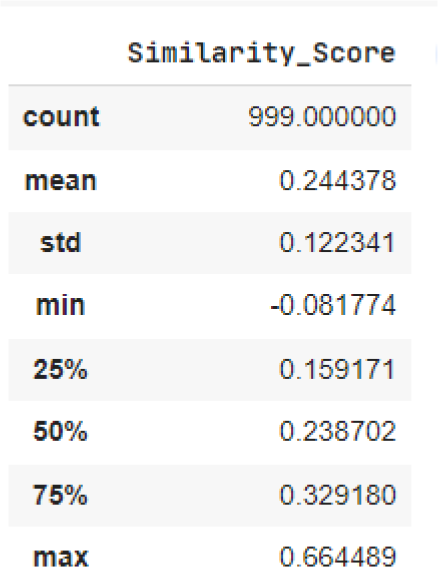
5.2 Cosine Similarity:

In our SIP model we checked cosine similarity of the similar features. “I love food” and “I like fruit” should give us the cosine similarity greater than 50 percentage so we will be able to understand that the two sentences are giving us the same meaning. But when we try to find the cosine similarity for the similar features like , we expected that the cosine similarity of all suicidal data will give us more than 50% similarity average but the number was pretty random .Thus, we could not rely on cosine similarity .



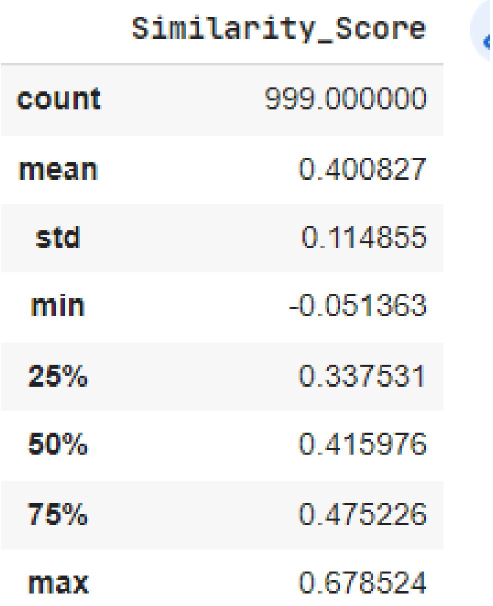
	Sentence	Similarity_Score
0	Am I weird I don't get affected by compliments...	0.077588
1	Finally 2020 is almost over... So I can never ...	0.226769
2	i need helpjust help me im crying so hard	0.324893
3	I’m so lostHello, my name is Adam (16) and I’v...	0.448409
4	Honetly idkI dont know what im even doing here...	0.418550

Figure 5.1: Cosine Similarity



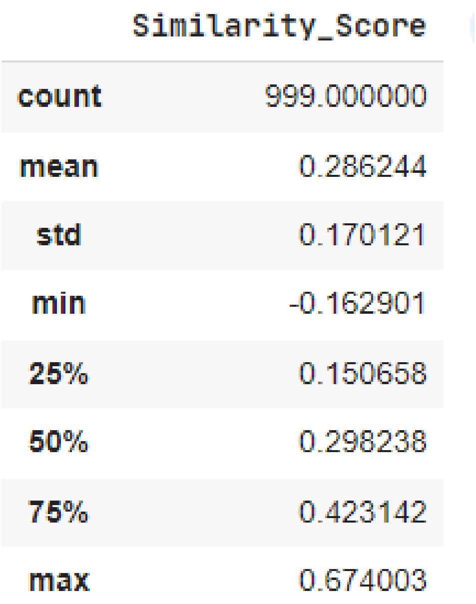
	Similarity_Score
count	999.000000
mean	0.244378
std	0.122341
min	-0.081774
25%	0.159171
50%	0.238702
75%	0.329180
max	0.664489

Figure 5.2: Suicidal



	Similarity_Score
count	999.000000
mean	0.400827
std	0.114855
min	-0.051363
25%	0.337531
50%	0.415976
75%	0.475226
max	0.678524

Figure 5.3: Non-Suicidal



	Similarity_Score
count	999.000000
mean	0.286244
std	0.170121
min	-0.162901
25%	0.150658
50%	0.298238
75%	0.423142
max	0.674003

Figure 5.4: Random

5.3 Results:

From our proposed model we get a training accuracy both training and validation accuracy of 89% SIP(Suicide) and from SIP(Sentiment) training accuracy of 98% and validation accuracy of 99%. Previously, we got highest 98% training and 67% validation accuracy from bi-directional LSTM, bi-directional GRU model and torch text models. There was over fitting issue at the time of training the Bi-LSTM and Bi-GRU Model.

Here are graphs of Training vs Validation accuracy:

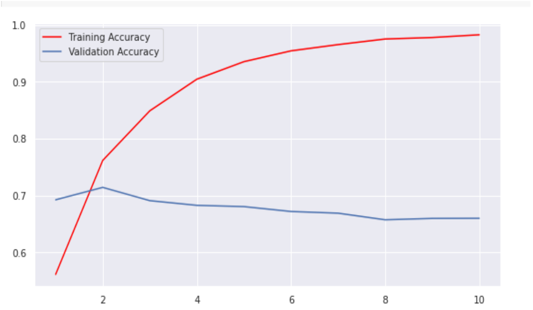


Figure 5.5: Bi-GRU

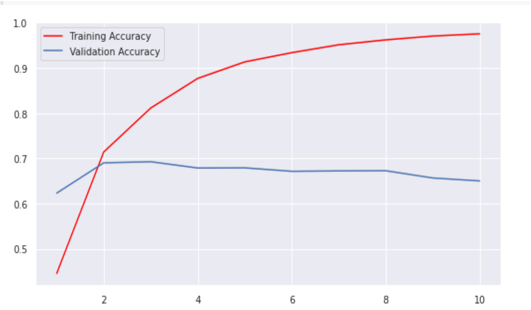


Figure 5.6: Bi-LSTM

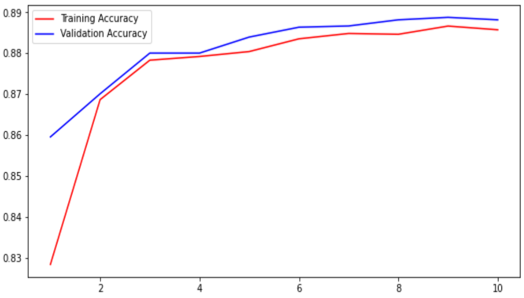


Figure 5.7: SIP (Suicide)

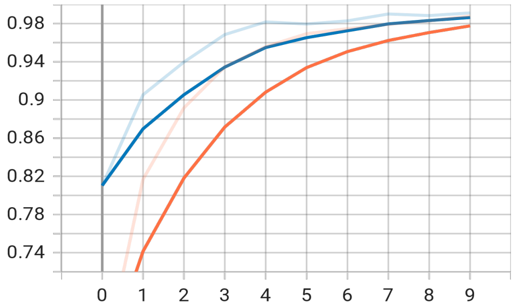


Figure 5.8: SIP

We got training accuracy 98% and validation accuracy 99% from SIP (Sentiment) model and the training accuracy was 88% and the validation accuracy 89% from the model where we used SIP(Suicidal).

Here are the graphs of Training vs Validation Loss:

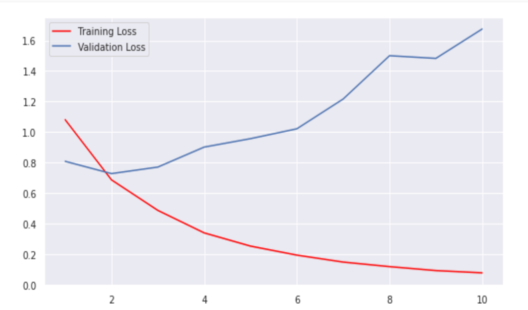


Figure 5.9: Bi-GRU

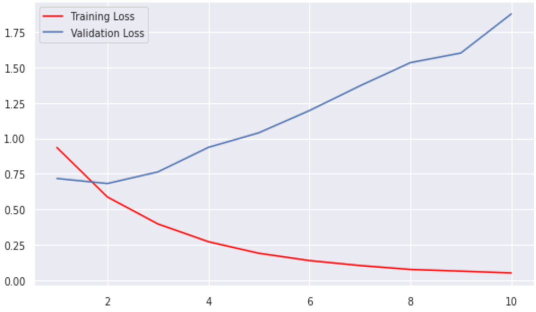


Figure 5.10: Bi-LSTM

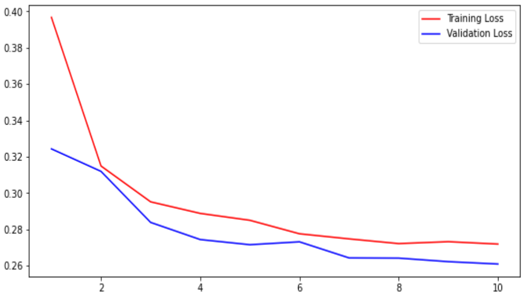


Figure 5.11: SIP (Suicide)

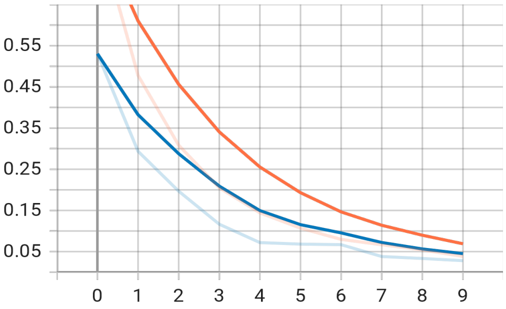


Figure 5.12: SIP(Sentiment)

The confusion matrixes we got from the SIP(Suicide) models from the two data sets are:

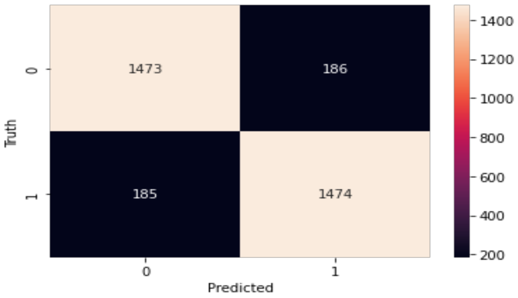


Figure 5.13: Dataset 01

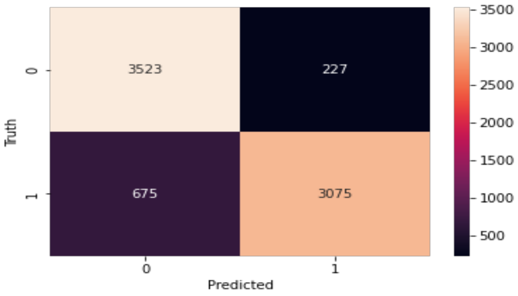


Figure 5.14: Dataset 02

We got a good training accuracy all the models Bi-LSTM, Bi-GRU, SIP but we have some over fitting issue while training Bi-LSTM and Bi-GRU that we could not solve by balancing dataset of dropout but still our proposed. Our SIP model solved that problem and gave more validation accuracy and precision then Bi-LSTM, Bi-GRU or Torchtext. We used two datasets for training, if we compare the results then we can see that the other model gave highest validation accuracy of 67% where SIP model gave highest validation accuracy of 99%.

Conclusion

Research regarding suicidal ideation detection using machine learning approaches based on social media text data is developing day by day. Nowadays, people are getting more connected and attached to social media. Social media has become the most popular way to express our feelings to connected people. The mental condition could easily be found in their text status on social media. As people share their feelings and daily life activities through their text status in social media, the current mental health is reflected through the thoughts they share. This is an effective way to determine whether someone has suicidal ideation or not. Analyzing the user social media data using machine learning approaches such as deep learning and artificial neural network could be an effective way to determine the ideation of a user. Finding out the problem using machine learning techniques is the main objective here. We have developed an algorithm that can quickly detect suicidal risk or ideation through social media data. We have followed three four methods such as our SIP model(BERT based), Bi-LSTM, Torchtext, and Bi-GRU, to bring out the differences such as which approaches are more capable of determining suicidal ideation with higher training accuracy 88% and with validation accuracy 89% and training accuracy 98% and validation accuracy 99% from SIP (Sentiment) model. We hope our proposed model will be able to determine suicidal ideation effectively, and the proper use of this model would help prevent suicidal cases in the future.

Bibliography

[1] L. Belfort, E. Mezzacappa, and E. Ginnis, “Similarities and differences among adolescents who communicate suicidality to others via electronic versus other means: A pilot study,” *Adolescent Psychiatrye*, vol. 2, no. 3, pp. 258–262, 2012.

[2] B. Dea, S. Wan, P. J. Batterham, A. L. Calear, C. Paris, and H. Christensen, “Detecting suicidality on twitter”,” *Internet Interventions*, vol. 2, no. 2, pp. 183–188, 2015.

[3] R. T. Liu, E. M. Kleiman, B. A. Nestor, and S. M. Cheek, “The hopelessness theory of depression: A quarter century in review”,” *Clin. Psychol*, vol. 22, pp. 345–365, 2015.

[4] H. Sueki, “The association of suicide-related twitter use with suicidal behaviour: A cross-sectional study of young internet users in japan”,” *Journal of Affective Disorders*, vol. 170, pp. 155–160, 2015.

[5] S. R. Braithwaite, C. Giraud-Carrier, J. West, M. D. Barnes, and C. L. Hanson, “Validating machine learning algorithms for twitter data against established measures of suicidality,” en, *JMIR Ment. Health*, vol. 3, no. 2, e21, May 2016.

[6] H. S. Bhat En and S. J. Goldman-Mellor, *Predicting Adolescent Suicide Attempts with Neural Networks”*. 2017.

[7] P. Burnap, G. Colombo, R. Amery, A. Hodorog, and J. Scourfield, “Multiclass machine classification of suicide-related communication on twitter,” en, *Online Soc. Netw. Media*, vol. 2, pp. 32–44, Aug. 2017.

[8] B. Dea, M. E. Larsen, P. J. Batterham, A. L. Calear, and H. Christensen, “A linguistic analysis of suicide-related twitter posts”,” *Crisis*, vol. 38, no. 5, pp. 319–329, 2017.

[9] M. Lapata, P. Blunsom, and A. Koller, *Reds, Proceedings of the 15th Conference of the European Chapter.* Valencia, Spain: Association for Computational Linguistics, 2017, vol. 1.

[10] A. E. Aladağ, S. Muderrisoglu, N. B. Akbas, O. Zahmacioglu, and H. O. Bingol, “Detecting suicidal ideation on forums: Proof-of-concept study”,” *J. Med. Internet Res*, vol. 20, no. 6, 2018.

[11] C. Berryman, C. J. Ferguson, and C. Negy, “Social media use and mental health among young adults”,” *Psychiatric Quarterly*, vol. 89, no. 2, pp. 307–314, 2018.

[12] J. Du, “Extracting psychiatric stressors for suicide from social media using deep learning,” *BMC Med. Inform. Decis. Mak*, vol. 18, no. S2, 2018.

[13] M. A. Franco-Martın, J. L. Muñoz-Sánchez, B. Sainz-de-Abajo, G. Castillo-Sánchez, S. Hamrioui, and I. de la Torre-Dıez, “A systematic literature review of technologies for suicidal behavior prevention,” en, *J. Med. Syst.*, vol. 42, no. 4, p. 71, Mar. 2018.

[14] S. Fodeh, T. Li, K. Menczynski, *et al.*, “Using machine learning algorithms to detect suicide risk factors on twitter,” in *2019 International Conference on Data Mining Workshops (ICDMW)*, Beijing, China: IEEE, Nov. 2019.

[15] I. Grandic, “How lstms work,” en, May 2019. [Online]. Available: https://izzygrandic.medium.com/how-lstms-work-263ac4e412ba.

[16] S. Jain, S. P. Narayan, R. K. Dewang, U. Bhartiya, N. Meena, and V. Kumar, “A machine learning based depression analysis and suicidal ideation detection system using questionnaires and twitter,” in *2019 IEEE Students Conference on Engineering and Systems (SCES)*, Allahabad, India: IEEE, May 2019.

[17] M. Morales, P. Dey, T. Theisen, D. Belitz, and N. Chernova, “An investigation of deep learning systems for suicide risk assessment”,” in *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology*, 2019, pp. 177–181.

[18] M. M. Tadesse, H. Lin, B. Xu, and L. Yang, “Detection of suicide ideation in social media forums using deep learning,” en, *Algorithms*, vol. 13, no. 1, p. 7, Dec. 2019.

[19] A. Roy, K. Nikolitch, R. McGinn, S. Jinah, W. Klement, and Z. A. Kaminsky, “A machine learning approach predicts future risk to suicidal ideation from social media data,” en, *NPJ Digit. Med.*, vol. 3, no. 1, Dec. 2020.

[20] S. Renjith, A. Abraham, S. B. Jyothi, L. Chandran, and J. Thomson, “An ensemble deep learning technique for detecting suicidal ideation from posts in social media platforms,” en, *J. King Saud Univ. - Comput. Inf. Sci.*, Nov. 2021.

[21] Y. Shaoxiong and S. Shiheng, “Clinical observation of laparoscopic anatomic hepatectomy for early primary liver cancer,” *E3S Web Conf.*, vol. 233, p. 02 026, 2021.