Suicide Detection from Ideation

Rohan M Lingeri
Dept. of Computer Science and Engineering
PES University
Bengaluru, India
rohanl.122002@gmail.com

Pratham Deepak Rao
Dept. of Computer Science and Engineering
PES University
Bengaluru, India
pratham.rao62@gmail.com

Rijul Saxena
Dept. of Computer Science and Engineering
PES University
Bengaluru, India
rijul.saxena@gmail.com

***Abstract*—Suicide is considered a low baseline behavior, which makes it challenging to predict. More than 700,000 people commit suicide each year. Reddit can be a good asset to detect the signs of suicide in a faster and more effective way thanks to the speed of information disseminated through it. By using machine learning and text analysis models, we examined how to predict suicide risk from written communications in an effort to advance suicide prediction and prevention. Specifically, we used a dataset consisting of more than 2,00,000 unique posts from "Suicide" and "Depression" subreddits on the Reddit Platform**

***Index Terms*—Suicide, Reddit, Depression, Machine Learning**

I. Introduction

Every year, almost 800,000 people commit suicide. Suicide remains the second leading cause of death among the young generation with an overall suicide rate of 10.5 per 100,000 people. According to predictions, there will be one death every 20 seconds by 2020 [1]. Almost 79% of the suicides occur in low- and middle-income countries where the resources for identification and management are often scarce and insufficient.

Suicide ideation is viewed as a tendency to end one's life ranging from depression, through a plan for a suicide attempt, to an intense preoccupation with self-destruction [2]. At-risk individuals can be recognized as suicide ideators (or planners) and suicide attempters (or completers). The relationship between these two categories is often a subject of discussion in research communities. According to some studies, most individuals with suicide ideation do not make suicide attempts. In WHO countries, early detection of suicide ideation has been developed and implemented as a national suicide prevention strategy to work towards the global market with the common aim to reduce suicide rates by 10% by 2020 [1].

Social media users, who are primarily young people, have developed a powerful "window" into their mental health and well-being in recent years. It offers anonymous participation in different cyber communities to provide a space for a public discussion about socially stigmatized topics.

II. Related Work

The authors in the paper [3] have used an NLP approach to detect tones of Depression in Reddit posts. The paper mainly focuses on deploying multiple feature extraction methods including N-grams, LIWC, and topic modeling using LDA, and testing their combinations to find the optimal result.

The authors use the results from feature extraction and plug them into a text classifier to estimate the likelihood of depression among the users using Logistic Regression, SVM, Random Forest, Adaptive Boosting, and Multilayer Perceptron classifier. The classification results are maintained in a confusion matrix to examine the effectiveness of the model.

Through experimentation, it is found that the combination of Bigrams and SVM produces the best results while using a singular feature extractor while a combination of all three aforementioned feature extractors along with a Multilayer Perceptron Classifier gave the best result overall.

The attention model was combined with the LSTM and CNN models to create a deep learning method for identifying suicidal intent from postings on social media sites. The model uses the resultant vector from an LSTM as an input value for the Attention layer and the convolution layer's output as an input. The LSTM-Attention-CNN combined model divides the input posts into four categories that indicate varying degrees of suicidal or non-suicidal intentions. The four categories are No Risk, Low Risk, Moderate Risk, and High Risk, respectively.

Every token or word in every phrase is mapped to a distinct index in the model's first layer, which creates a real-valued vector of a specific length. A dropout layer is used to prevent over-fitting. The TF-IDF Library contains the TF-IDF Vectorizer's basic code, which was examined. The most frequent words were gathered, and then TF-IDF was used to give each word a weight. The net weight of a given post was then examined using these weights. Threshold values that divide each post into its corresponding labels were chosen after several experiments and testing.

Over the textual input, the LSTM layer detects long-distance relationships, and the convolutional layer, which aids in feature extraction, is used to extract the features. The attention model emphasizes crucial information and gives each word weight. The feature map is down-sampled by the pooling layer, which lists all the features that are present in a certain area. The information gathered by the pooling layer will be transformed into a column vector by the flattened layer. The output layer then uses the SoftMax algorithm to classify the input posts.

[5] used a dataset that consisted of 27,329 texts with lengths ranging from 50 to 20K characters. Out of this 193 were genuine notes of people who committed suicide. The linguistic Indicator Approach was used to identify a set of indicators that can be used to detect suicidal ideation. A dictionary-based approach was used, each dictionary contained a set of words that represented an indicator, then the relative frequencies of the words in the text material were counted. The prevalence of each word was taken by taking the occurrence of each word divided by the total number of words to obtain a score for each variable.

Machine Learning Approach: SVM(Support Vector Machines) and RoBERTa(Robustly Optimized BERT Pre-training Approach). SVM was used to convert all the letters into lower cases. TF-IDF((Term Frequency–Inverse Document Frequency) was used to denote how important a word is in the collection of texts. While classical machine learning approaches, such as SVM, made use of a bag of words (BOW) to create numerical features, the sequential order of the words and their relations were not considered.

A RoBERTa tokenizer was applied, which has the rules to tokenize text, as well as the vocabulary and dictionary mapping tokens to numerical indices. The maximum sequence of tokens was fixed to 512 tokens, and the Adam optimizer was chosen. The experiment concluded with 2 epochs and a batch size of 8.

III. Problem Formulation

Due to the cover of anonymity provided by social media, more and more people have taken to expressing their thoughts on social media, without great risks to their identity. Naturally, these channels have also become the medium for last-ditch communications for people who wish to share their emotions, pain, and suicidal thoughts, and are at high risk of becoming victims of suicide. This proves as a really important source for identifying high-risk situations pertaining to suicide before the victims end their life. Early detection and treatment are the best ways to prevent suicide attempts.

Potential suicide victims might have brief thoughts or make a suicide plot to convey their suicidal tendencies on the internet. To identify these risky intentions or actions before it becomes irreversible, suicidal ideation detection is used. These detection methods most often tap into the aforementioned sources of information on the internet.

It is imperative to identify and report occurrences of suicidal posts on the internet which could end up saving a life given proper attention. A quick look at a social media platform hosting such conversations as r/suicidewatch on the Reddit platform gives us insights into how important time is in dealing with these situations. Posts indicating a threat to a user's life within hours of writing a final message are far too common of an occurrence.

One of the challenges faced in solving the problem of detection of suicidal tendencies from online messages is accurately differentiating between posts that are related to mental illnesses and suicide. From exploratory analysis, one can conclude that both mental illness-related posts and suicide-related posts share an overlapping word base. Another task in the proposed solution is to find correlations between terms in the analyzed messages so conclusive decisions can be made based on historical data. The important task in creating a solution for the given situation is to maximize the true positives, be it at the cost of higher false positives, i.e. a model that classifies all truly suicidal text as suicidal with leeway in classifying non-suicidal posts as suicidal.

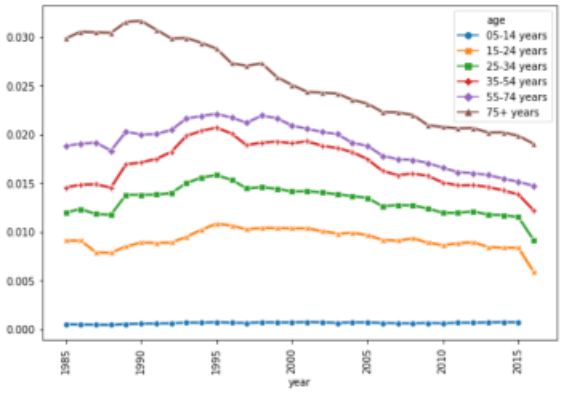
Our proposed solution involves experimenting with different methods of sentiment analysis, including feature extraction, and different classification methods to define if a text indicates suicidal tendencies. Our solution also aims to incorporate the classification of the given text into different sub-categories based on the plausible cause of the situation of the writer (financial, social, personal, etc.) so that targetted help can be provided to the person

IV. Data

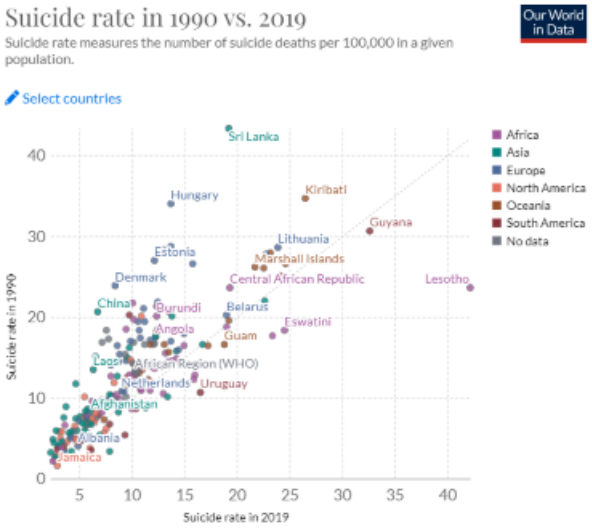
A. Statistics of Data

One of the foremost challenges in the domain of suicidal ideation detection is the lack of availability of a public dataset due to privacy and anonymity concerns borne out of social stigma associated with mental illness and suicide. We train our classification model with reddit dataset where users express their views and opinions via submissions. They interact through comment threads attached with every submission or post of the different users. The dataset we used is a collection of posts from "SuicideWatch" and "depression" subreddits of the Reddit platform. The posts are collected using Pushshift API. All posts that were made to "SuicideWatch" from Dec 16, 2008(creation) till Jan 2, 2021, were collected while "depression" posts were collected from Jan 1, 2009, to Jan 2, 2021.

After having the complete dataset, we decided to do some research regarding the statistics of our data. According to the data and insights provided by [6] most of the suicide cases that occurred were in the age group of 15-49 years old while the least cases were seen in the age group 5-14 years old.



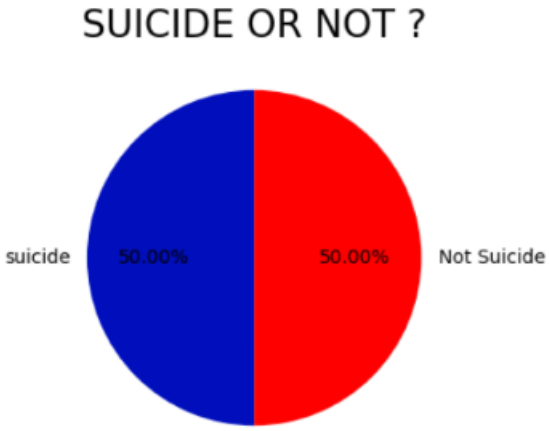
We also found out that over the years, there was a drastic decline in the number of suicides since 1990 to 2019. Given below, we see the overall picture is mixed: the majority of countries lie above the grey line, meaning suicide rates have fallen since 1990. But a significant number fall below it, indicating an increase over this period. Most countries in Europe have seen a decline in suicide rates; Asia too has seen impressive declines. Across other regions, the trend has been more varied.



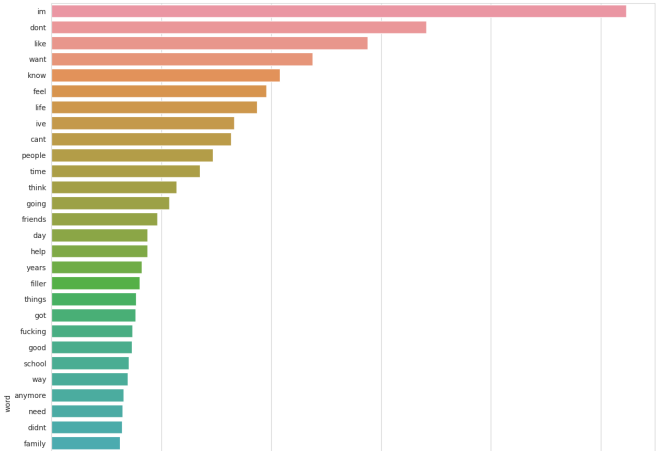
Suicide is more common in men than women in all countries Globally, suicide rates in men are just over twice as high as for women. In 2017 – as we see in the visualization – the global suicide rate for women was 6.3 deaths per 100,000; for men, it was just over twice that figure at 13.9 per 100,000. The difference in rates for males and females can be explored for all the continents in the image given below.

B. *Exploratory Data Analysis*

A significant part of the project involved preprocessing the data in order to bring it to a suitable form that could then be subjected to models. The raw dataset considered of multiple Comma-Separated Values (CSV) files that had to be joined and filtered. After analysing the data, we understood that our data consisted of 2,32,074 distinct texts. Out of which 1,16,037 which is 50% was classified as suicide and the rest 50% of the data was classified as non-suicide. Given below is the pie diagram stating how our data is classified into 2 classes.



Followed by the analysis of classes, we advanced with data cleaning. Initially, all the stopwords and null values (if any) were removed and all the special characters were grouped. For cleaning of data, an module called 'neattext' was used which provided us with readily available functions to separate data into trained clean text and test clean text. Using a barplot we inspected which special characters occurred most frequently.



V. Methodology

A. *Data pre-processing*

We use various methods to clean the data to make it ready for feature extraction. The data is normalized by making all characters lowercase and removing special characters. Insignificant stop words are removed from the data.

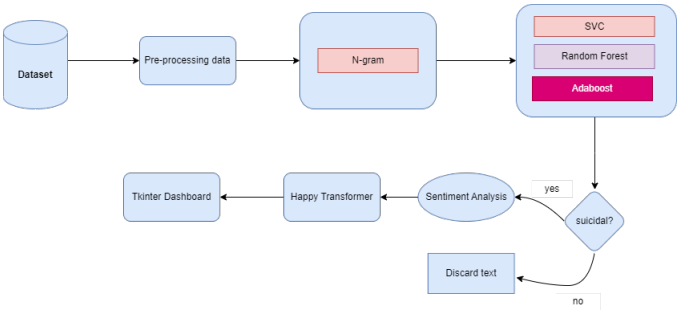


Fig. 1. Flowchart of methodology

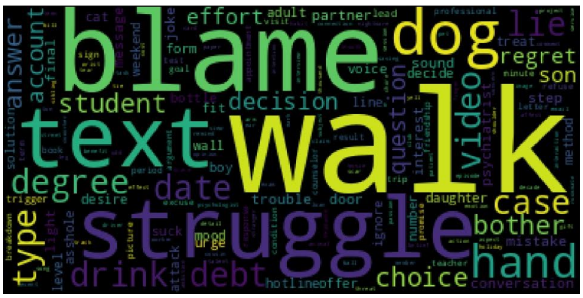


Fig. 2. Most frequent n-grams in suicide-related posts3

Words with vaguely similar meanings (such as open, opening, etc) are stemmed down to singular words (open) so there is lower word variance.

B. Feature Extraction and Encoding

After data pre-processing, we feed features of the text/users learned from the provided data. These linguistic habits are achieved by extracting n-grams. N-grams formed on provided data give information about the most important features which in our case include unigrams (singular words) and bigrams (two words that hold more meaning together). For N-grams we use the pre-trained Phrases model to discard model states that are not strictly needed for the phrase detection task. We give more importance to n-grams with more than 3 occurrences in the document. The most frequent n-grams in the dataset considered for each category have been shown in Fig 1. and Fig 2.

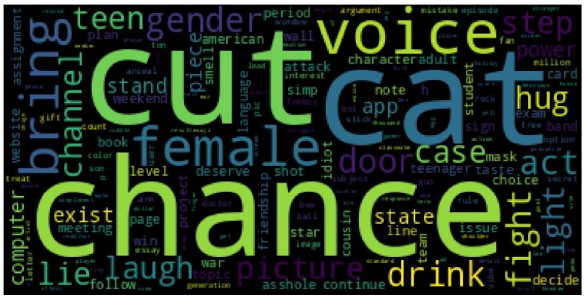


Fig. 3. Most frequent n-grams in non-suicide-related posts3

N-gram	No. of occurances
die	40517
suicide	45743
helpless	684
hope	20530
feel worse	1333
feel like	44361
dont know	42911

TABLE I
FEW N-GRAMS AND THE NUMBER OF OCCURRENCES IN THE DATA

Converting words to vectors can be achieved in multiple ways. TF-IDF (Term Frequency Inverse Document Frequency) ideates vectors for individual words based on the term frequency of all the words in the document. However, to achieve a deployable product capable of converting isolated occurrences of words in the form of scraped input, a trainable vectorizer is required. Our initial solution incorporated TF-IDF as the vectorizer which was ideal for a train - test evaluation environment. However as our solution demanded vectorization of text without past context being available, we had to choose another method for embedding.

The Word2vec model [9] is used to remember encodings for each word in the cleaned data. Word2vec is a two-layer neural network that processes text by vectorizing words. A text corpus serves as its input, and its output is a collection of feature vectors, which stand in for the words in the input corpus. It turns text into numerical data which learning models can understand. Word2vec can develop extremely accurate assumptions about a word's meaning based on prior occurrences if given enough information, usage, and contexts.

The model is trained on the normalized training data. Each n-gram is then encoded into vectors with 50 information dimensions based on the information gained during the training phase. The vectors then produced locate each word as a point in 50-dimensional vectorspace. This is done for each word in each sentence to produce a vector representation of the dataset. Similar ideas and things can be measured as 'close' using the vectors as position vectors.

As each sentence is of varying length (words), the vectors achieved from the word2vec model are not of a uniform size. However, as the training model expects a uniform sized input, to retain as much data as possible while increasing or decreasing dimensions, we have adopted the method of averaging the vectors to get the same dimensions for each data instance.

C. Classification

After the data has been embedded into model understandable vectors, we pass the vectors into the classification models.

1) SVM: Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used

for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. However, a major drawback of SVMs is that they become incredibly complex, increasing training time and decreasing accuracy when the number of data points is more than 10 thousand. Given that our dataset has 180 thousand and more data points, we don not use SVMs in the final model.

2) Random Forest: Random Forrest is an ensemble learning method involving the training of multiple decision trees on the training data. The classification result produced is the result determined by the most number of trees in the model. Random forest is better suited than SVMs in handling high volumes of data. We chose the random forest model for further use.

3) AdaBoost: AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. In some problems it can be less susceptible to the overfitting problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner.

4) MLP: A Multilayer Perceptron has input and output layers, and one or more hidden layers with many neurons stacked together. And while in the Perceptron the neuron must have an activation function that imposes a threshold, like ReLU or sigmoid, neurons in a Multilayer Perceptron can use any arbitrary activation function. Multilayer Perceptron falls under the category of feedforward algorithms, because inputs are combined with the initial weights in a weighted sum and subjected to the activation function, just like in the Perceptron. But the difference is that each linear combination is propagated to the next layer.

D. Sentiment Analysis

To assess the severity of a text classified as a potential suicide risk, sentiment analysis is done. We have chosen the pretrained sentiment classifying algorithm Happy Transformer [10]. The transformer employs the transformer model for context-based textual analysis. The text classification model returns the severity value of the provided text.

VI. Experimental results

Upon testing the trained models on test dataset, we achieved various results for each of them. The fitness parameters chose by us are accuracy, f-1 score, precision and recall. Accuracy measures the proportion of correctly made classifications to total number of classifications. Precision gives the exactness of our model in the positively predicted classifications. Recall gives us the fraction of items which are not labelled as belonging to the positive class but should have been in a positive classification. F1-score is the harmonic mean between precision and recall, it signifies the mean of an individual's performance.

Performance results of classification models				
Fitness measure	SVC	RF	Adaboost	MLP
Accuracy	74.20%	76.82%	74.27%	78.71%
F-1 score	0.7420	0.7682	0.7427	0.7871
Precision	0.7328	0.7850	0.7342	0.7894
Recall	0.7411	0.7387	0.7608	0.7832

TABLE II
PERFORMANCE METRICS FOR USED MODELS

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

Based on the results shown in table 2, the best results is obtained from the MLP model hence we use it in our final solution

VII. Conclusion and future work

In this paper we tried to scrape suicidal data posted on the Reddit social media platform. We cleaned the text, vectorized it and classified if it indicates suicidal tendencies using the models Word2vec for embedding and MLP for classification. The MLP model outperforms SVC, Random Forest and Adaboost models and provides an accuracy of 78.71%. If classified as suicidal, the text is then sent to the Happy Transformer sentiment analysis model to get a numeric figure between 0 and 1 to indicate the severity. This data is then stored and can be viewed using a basic frontend.

Although the model accuracies are satisfactory, the low values indicate it is a challenging task and has room for improvment. Using positional contexts while vectorizing the text using transformers can also imporve results as the model will have a better understanding of finding and training of vectors.

Acknowledgment

We would like to express our special gratitude to our beloved teacher Mr. Srinivas Kathargуppe who gave us this wonderful oppurtunity to work on this amazing project. We came to know about a lot of new things and understood the workflow of how research works in the real world which we are really thankful for.
Secondly we would also like to express our heartfelt appreciation to college for providing us with the necessary resources

when and where required, which played a major role in the successful completion of the project.

REFERENCES

[1] World Health Organization. National Suicide Prevention Strategies: Progress, Examples and Indicators; World Health Organization: Geneva, Switzerland, 2018.

[2] Beck, A.T.; Kovacs, M.; Weissman, A. Hopelessness and suicidal behavior: An overview. JAMA 1975, 234, 1146–1149.

[3] M. M. Tadesse, H. Lin, B. Xu and L. Yang, "Detection of Depression-Related Posts in Reddit Social Media Forum," in IEEE Access, vol. 7, pp. 44883-44893, 2019, doi: 10.1109/ACCESS.2019.2909180.

[4] Shini Renjith, Annie Abraham, Surya B. Jyothi, Lekshmi Chandran, Jincy Thomson, An ensemble deep learning technique for detecting suicidal ideation from posts in social media platforms, Journal of King Saud University - Computer and Information Sciences, 2021,https://doi.org/10.1016/j.jksuci.2021.11.010.

[5] A. Shrestha, N. Akrami, L. Kaati, J. Kupper and M. R. Schumacher, "Words of Suicide: Identifying Suicidal Risk in Written Communications," 2021 IEEE International Conference on Big Data (Big Data), 2021, pp. 2144-2150, doi: 10.1109/BigData52589.2021.9671472.

[6] https://ourworldindata.org/suicide

[7] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[8] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[9] L. Ma and Y. Zhang, "Using Word2Vec to process big text data," 2015 IEEE International Conference on Big Data (Big Data), 2015, pp. 2895-2897, doi: 10.1109/BigData.2015.7364114.

[10] A. K. J, E. Cambria and T. E. Trueman, "Transformer-Based Bidirectional Encoder Representations for Emotion Detection from Text," 2021 IEEE Symposium Series on Computational Intelligence (SSCI), 2021, pp. 1-6, doi: 10.1109/SSCI50451.2021.9660152.