Suicidal Thoughts Detection from Social Media Using AI

By Rishav Biswas

Author Bio

Rishav Biswas is a high school graduate from Kolkata, India. He is 18 years old and passionate about AI. Three months ago, he got rejected from every college he applied to, including the safety schools also. He wants all the readers to always dream big but also remember that it will hurt a lot when you fail. Even if it hurts, you will be living with the satisfaction of trying. Through this research, he got to learn so many things which are not taught in his school. He has a dream of reducing the suicide rate in this world.

Abstract

Suicide is one of the leading causes of death worldwide. Early detection and prevention of suicide attempts should be addressed to save lives. Nowadays, people on social media are posting posts that include suicidal thoughts. To detect this kind of post, the author has used an AI model, which can do this by analyzing text. This study is about how the model works and the results of testing the model. The study will cover every aspect of the model. The study also includes an explanation of GloVe, which is used for word embedding. The author has also discussed the weaknesses of the model. There are some pre-existing models for suicidal thoughts detection, but their accuracies are not as high as this model. This model can detect suicidal thoughts with a recall of 0.93 (93 percent) and a precision of 0.94 (94 percent). The author thanks Dr Ganesh Mani, an instructor at Carnegie Mellon University (and editorial board member recused from the review of this paper), and Alfred Renaud for helping him to complete this research project.

Keywords: Machine Learning, AI modeling, suicide, suicidal thoughts, suicide detection, social media, AI, suicidal ideation, suicidal thoughts detection

Introduction

Suicide is among the top three causes of death among youth worldwide. According to the WHO (5. World Health Organization, https://www.who.int/news-room/fact-sheets/detail/suicide), every year, almost one million people die from suicide, and 20 times more people attempt suicide; a global mortality rate of 16 per 100,000, or one death every 40 seconds and one attempt every 3 seconds, on average. The rates of suicide have greatly increased among youth, and youth are now the group at highest risk in one-third of the developed and developing countries.

Due to the advances of social media and online anonymity, an increasing number of individuals turn to interact with others on the internet. Online communication channels are becoming a new way for people to express their feelings, suffering, and suicidal tendencies. Hence, online channels have naturally started to act as surveillance tools for suicidal ideation, and mining social content can improve suicide prevention. In addition, strange social phenomena are emerging, e.g., online communities reaching an agreement on self-mutilation and copycat suicide. For example, a social network phenomenon called the "Blue Whale Game" in 2016 allegedly used many tasks (such as self-harming) and allegedly influenced game members to commit suicide in the end (6. for a journalistic review of the phenomenon, see BBC, https://www.bbc.co.uk/news/blogs-trending-46505722). Suicide is a critical social issue and takes thousands of lives every year. Thus, it is necessary to detect suicidality and prevent suicide before victims end their life. Early detection and treatment are regarded as the most effective ways to prevent potential suicide attempts.

Social media has become an integral part of our lives, especially for the youth. Many people use it to share photos and funny stories, talk about their political viewpoints and catch up with friends. It can also be a place where they feel comfortable talking about struggles or their pain. It may be easier to write about how they feel behind a screen than to share it with someone in person. When they are struggling, they may isolate themselves, and social media becomes one of their only connections to others.

There has been an increase in the number of posts where people are talking about emotional problems. These posts sometimes go one step further where people are expressing their suicidal thoughts. There is a chance that these people may be involved in a suicidal attempt in the future. If I can detect these types of posts, it may help in suicide prevention.

Now to detect suicidal thoughts, I have to take the help of an AI model. The AI model is used for the training and deployment of machine learning algorithms that emulate logical decision-making based on available data. To build an AI model, I need a dataset that can be used for training and testing its purpose. The text involved in a post is the primary source of data to be used as a data set.

In this study, I have taken the dataset from subreddits on the Reddit platform. I have used GloVe for word embedding [3]. In natural language processing, word embedding is a term used for the representation of words for text analysis, typically in the form of a real-valued vector that encodes the meaning of the word such that the words that are closer in the vector space are expected to be similar in meaning. Then the model was created to detect suicidal thoughts.

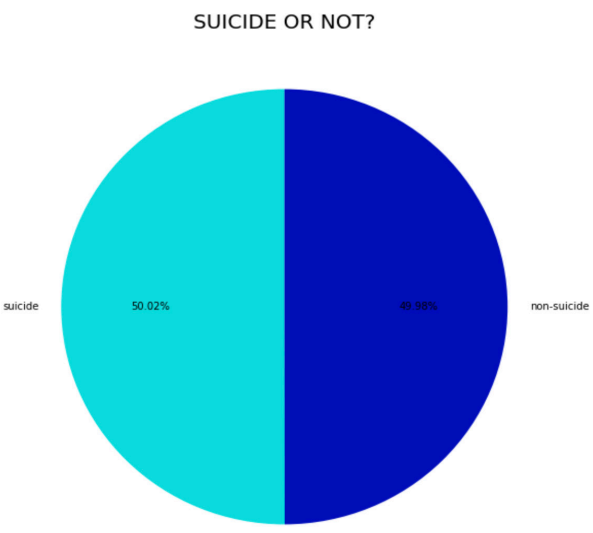
Methods

Data Collection and Preparation

I have collected the data from (https://www.kaggle.com/datasets/nikhileswarkomati/suicide-watch). The dataset consists of 232,074 posts, which are classified into two classes. The two classes are suicide and non-suicide. The dataset is a collection of posts from "SuicideWatch" and "depression" subreddits on 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. The dataset is illustrated in the table below.

#	TEXT	CLASS
29	yea putting a knife to my wrist didn't give me any hesitation like how it used to, i am free from tha...	Suicide
30	I am ending my life today, goodbye everyone.I am 36 almost 37, I am on disability for PTSD and Rheum...	Suicide
37	Guys I want friends That's it , I'm alone and don't talk to anyone dm me or anything, I'm just tired...	non-suicide
4	Finally 2020 is almost over... So I can never hear "2020 has been a bad year" ever again. I swear to...	non-suicide
25	I'm scared. Everything just seems to be getting worse and worse. I'm young and I think I'm transge...	Suicide

In the above table, there are three columns: the first column is unique values assigned to every text, the second column is text, and the third column is about the class in which a particular text falls. To demonstrate the percentage of text that fall into two classes, there is a pie chart below.



The two classes have almost the same number of data, around 50 % each. This is important for any kind of biased results for my model. There are some changes made to the original dataset due to mismatches of data. This is quite significant for the accuracy of my model.

The dataset was split into two subsets for training and testing of the model. Then, I cleaned the text data by removing special characters or any stopwords (There are a lot of commonly used words, such as 'the', 'is', 'that', 'a', etc., that would completely dominate an analysis, but don't offer much insight into the text in the documents; these words that I want to filter out before analysing the text are called 'stopwords') and converting it to lower case. After that, it was divided into tokens by the process of tokenization (Many TDM methods are based on counting words or short phrases. However, a computer doesn't know what words or phrases are – to it, the texts in your corpus are just long strings of characters. You need to tell the computer how to split the text up into meaningful segments that will enable it to count and perform calculations. These segments are called tokens, and the process of splitting your text is called tokenization).

For pre-processing the text data, I have used the pad_sequence() function from the Keras library [2]. It is done to ensure that all inputs are of the same size. Then I used Label Encoder, from the scikit-learn library, to convert labels (words of the text are labeled) into a numeric form to convert them into the machine-readable form. It will help the model to decide in a better way how those labels must be used.
The data is now collected and prepared for the model to be used.

Word Embedding and GloVe

Word Embedding is the process of converting high-dimensional data to low-dimensional data in the form of a vector in such a way that the two are semantically similar. In its literal sense, "embedding" refers to an extract (portion) of anything. Generally, embeddings improve the efficiency and usability of machine learning models and can be utilized with other types of models as well. When dealing with massive amounts of data to train, building machine learning models is a nuisance. As a result, embedding comes into play.

I have used GloVe for word embedding. GloVe stands for Global Vectors [3]. The GloVe is

essentially a log-bilinear model with a weighted least-squares objective. The main intuition underlying the model is the simple observation that ratios of word-word co-occurrence probabilities have the potential for encoding some form of meaning. For example, consider the co-occurrence probabilities for target words 'ice' and 'steam' with various probe words from the vocabulary. Here are some actual probabilities from a 6-billion-word corpus:

Probability and Ratio	K = Solid	K= Gas	k = Water	K = fashion
P(K|ice)	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
P(K|steam)	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
P(K|ice)/P(K|steam)	8.9	8.5×10^{-2}	1.36	0.96

As one might expect, ice co-occurs more frequently with solids than it does with gas, whereas steam co-occurs more frequently with gas than it does with solids. Both words co-occur with their shared property, water frequently, and both co-occur with the unrelated word fashion infrequently. Only in the ratio of probabilities does noise from non-discriminative words like water and fashion cancel out so that large values (much greater than 1) correlate well with properties specific to ice, and small values (much less than 1) correlate well with properties specific to steam. In this way, the ratio of probabilities encodes some crude form of meaning associated with the abstract concept of the thermodynamic phase.

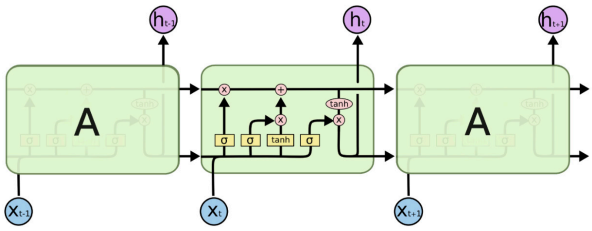
The training objective of GloVe is to learn word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence. Since the logarithm of a ratio equals the difference of logarithms, this objective associates (the logarithm of) ratios of co-occurrence probabilities with vector differences in the word vector space. Because these ratios can encode some form of meaning, this information gets encoded as vector differences as well.

I have used a pre-trained GloVe Embedding taken from(https://www.kaggle.com/datasets/authman/pickled-glove840b300d-for-10sec-loading) to build my model. It was of size 2.3 Gb. It was trained on a dataset of 84 billion tokens(words) with a vocabulary of 2.2 million words and an embedding vector size of 300 dimensions. I can seed the Keras (a library) Embedding layer with weights from the pre-trained embedding for the words in the training dataset. Then I can define the examples, encode them as integers, and then pad the sequence of similar length. Keras provides a Tokenizer class that can be fit on the training data, can convert text to sequences consistently by calling the texts_to_sequences() method on the Tokenizer class, and provides access to the dictionary mapping of words to integers in a word_index attribute.

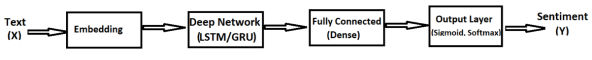
I need to load the entire GloVe word embedding file into memory as a dictionary of words for the embedding array. After that, I need to create a matrix of one embedding for each word in the training dataset. I can do that by enumerating all unique words in the Tokenizer.word_index and locating the embedding weight vector from the loaded GloVe embedding. I will be using this embedding_matrix in my model.

Model Training and Working

I want to build a sequential model with the help of the Recurrent Neural Network (RNN) method. Sequence models are machine learning models that input or output sequences of data. Keras is the main Library that I have used to build this model [2]. To use the RNN method, I have used a built-in RNN layer of Keras (a library) named LSTM(Long short-term memory).



In the above picture, it is a typical LSTM unit that is repeated over the whole length of a sequence. In my model, I first created layers, and with the help of LSTM and embedded_matrix, which was already created, I can train my model. The model performed 20 epochs with a batch_size = 256 on the training dataset. After that, for testing, I used the testing data set. This data set was already split earlier.



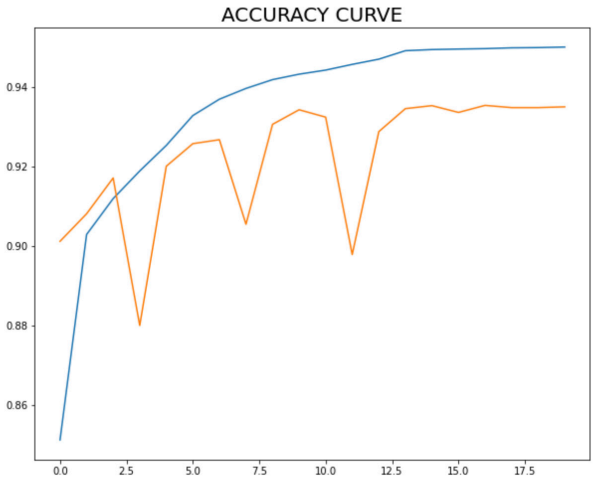
From the above picture, it can be understandable how my model works. At first, there

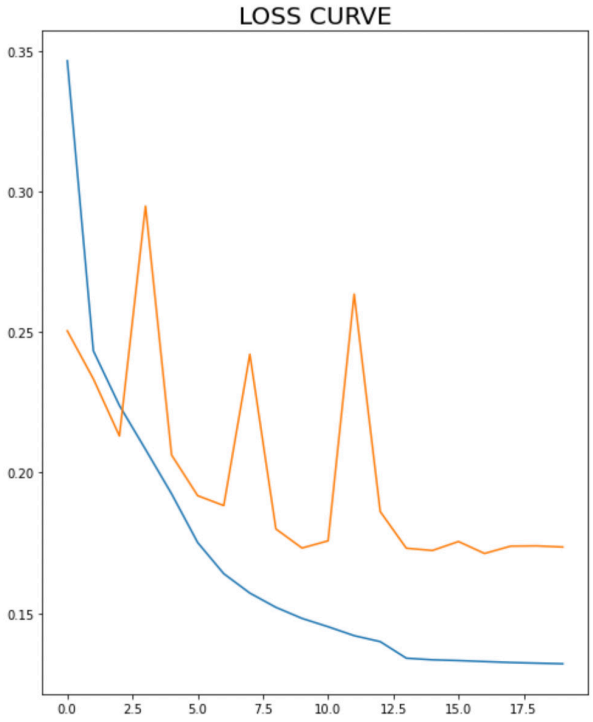
is the text which is the input data, and then there is an embedding layer in which it uses word embedding. Thirdly it has a Deep network (LSTM) that takes the sequence of embedding vectors as input and converts them to a compressed representation. The compressed representation effectively captures all the information in the sequence of words in the text. Fourthly, the fully connected layer takes the deep representation from the LSTM and transforms it into the final output classes. Lastly, it displays its sentiment as either suicide or non-suicide.

Results

In the below graph, the blue line is the accuracy of the training set, and the orange line is the validation set accuracy. The x-axis is for accuracy, and the y-axis is for epoch.

The Model has achieved a weighted average precision of 0.95(95 percent), recall of 0.95(95 percent), and f1-score of 0.95(95 percent) on the training set. For the testing set, the precision, recall, and f1-score are 0.94(94 percent), 0.93(93 percent), and 0.93(93 percent) respectively. The precision for the suicide classes is high than the non-suicide classes for the testing set, while the recall is just the opposite. In the training set, both the precision and recall are the same.





In the above graph, the blue line is the accuracy of the training set, and the orange line is the validation set accuracy. The x-axis is for loss, and the y-axis is for epoch.

Discussion

This study gives a clear picture of how a model is created and how it works. The accuracy of the model is slightly high in comparison to other works on suicidal thought detection. There was a significant change in accuracy when I made some changes to the dataset. My next step will be to analyze the dataset for more scope for improvement in accuracy.

I will be using more datasets in the future to test and improve the accuracy of this model. From the results, it is unclear why there is a difference between precision and recall. I will be looking into that. There will always be uncertainty about the reality of the text means somebody may be for fun posting suicidal thoughts or sarcastically doing this. For these reasons, I should focus on future studies to make a more efficient model. I will also be working on how to reduce the loss of the model during its work.

The ultimate aim of suicidal thoughts detection is intervention and prevention. One of the applications of this study is Proactive Conversational Intervention. Very little work is undertaken to enable proactive intervention. Proactive Suicide Prevention Online (PSPO) [4] provides a new perspective with the combination of suicidal identification and crisis management.

An effective way is through conversations. For enabling timely intervention, automatic response generation becomes a promising technical solution. Natural language generation techniques can be utilized for generating counseling responses to comfort people's depression or suicidal ideation. Reinforcement learning can also be applied to conversational suicide intervention. After suicide attempters post suicide messages (as the initial state), online volunteers and lay, individuals will take action to comment on the original posts and persuade attempters to give up their suicidality.

Conclusion

The study is successfully able to interpret the model. The study also gives every detail about the model building, which will be helpful to beginner students. The model has an accuracy of around 94 percent, which can further be increased. Overall, if you look into the study, it is quite evident that AI can successfully detect suicidal thoughts from social media. Online social content is very likely to be the main channel for suicidal ideation detection in the future [1]. It is, therefore, essential to develop new methods which can heal the schism between clinical mental health detection and automatic machine detection, to detect online texts containing suicidal ideation in the hope that suicide can be prevented.

References

1. Ji, S., Pan, S., Li, X., Cambria, E., Long, G., & Huang, Z. (2021). Suicidal ideation detection: A review of machine learning methods and applications. IEEE Transactions on Computational Social Systems, 8, 214–226.

2. Joseph, F., Nonsiri, S., & Monsakul, Annop. (2021). Keras and TensorFlow: A hands-on experience. In Prakash, K., Kannan, R., Alexander, S., Kanagachidambaresan, G. (Eds.), Advanced deep learning for engineers and scientists (pp. 85–111). Springer, Cham. http://dx.doi.org/10.1007/978-3-030-66519-7_4

3. Pennington, J., & Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. Proceedings of the Conference on Empirical Methods in Natural Language Processing, 14, 1532–1543. http://dx.doi.org/10.3115/v1/D14-1162

4. X. Liu, X. Liu, J. Sun, N. X. Yu, B. Sun, Q. Li, and T. Zhu, "Proactive suicide prevention online (pspo): Machine identification and crisis management for Chinese social media users with suicidal thoughts and behaviors," Journal of Medical Internet Research, vol. 21, no. 5, p. e11705, 2019.

5. World Health Organization, https://www.who.int/news-room/fact-sheets/detail/suicide
6. BBC, https://www.bbc.co.uk/news/blogs-trending-46505722