Suicidal text detection on social media for suicide prevention using deep learning models

Neha Sharma[1] and Prashant Karwasra[2]
School of Computing
Indian Institute of Information Technology, Una, India
[1]neha724@gmail.com, [2]karwasraprashant10@gmail.com

Abstract- **The advent of social media has transformed the way we communicate and connect, enabling individuals worldwide to instantly and openly interact with friends, family, and colleagues on a frequent basis. People utilize social media platforms as a means to express their opinions, share personal experiences, narratives, and challenges. Nevertheless, concerns have arisen due to the growing prevalence of suicidal content on social media platforms, where discussions of hardship, thoughts of death, and self-harm are widespread, particularly among younger generations. Consequently, harnessing the power of social media to detect and identify suicidal behavior, including the presence of suicidal thoughts, becomes essential in offering appropriate interventions that discourage self-harm and suicide, as well as in preventing the spread of suicidal ideations throughout these platforms. This paper presents suicidal content detection using two deep learning architectures, LSTM, and DistilBERT with the latter showing better performance in respectively. We conclude by drawing implications for deep learning architectures in detecting suicidal content on social media and an initial deployment of the models using Telegram bot which detects the message containing suicidal content and sends a motivational message in response and also informs their friends and relative through alerts.**

Keywords: *social media, suicide, suicidal-content, suicidal ideation, LSTM, DistilBERT, Deep Learning.*

I. INTRODUCTION

Annually, approximately 703,000 individuals succumb to suicide, translating to a distressing frequency of one person per second, with many more individuals attempting it [1]. The impact of each suicide is profound, affecting not only families and communities but entire nations, leaving lasting consequences on those left behind. It is important to recognize that suicide spans across all age groups, and in 2019, it ranked as the fourth leading cause of death among individuals aged 15-29 worldwide [1]. Suicide denotes a deliberate act wherein an individual intentionally seeks to end their own life [2]. The act of suicide is a multidimensional phenomenon that arises from the intricate interplay of various factors, including biological, psychological, social, cultural, and spiritual elements [3]. Numerous individuals opt for social media platforms as a means to express their thoughts, emotions, and everyday encounters, including the challenges and difficulties they face. Suicidal ideation, thoughts of death, and self-harm emerge as prevalent topics in these online discussions. The identification of individuals expressing suicidal tendencies or harboring suicidal thoughts through their tweets or blogs holds immense significance. This is because the early detection of such individuals has the potential to prevent numerous lives from being lost. In this paper we have examined social media content to detect automatically suicidal ideation and behaviors with the help of LSTM and DistilBERT model architectures. This paper presents a detailed overview of above two deep learning architectures that can be used for detection of suicidal content and identify the suicidal ideation. However, little is known on how some deep learning models account for detecting suicidal text on social media. Performing this investigation is the main goal of this paper.

II. SOCIAL MEDIA DATASET

The social media dataset, curated by Nikhileswar Komati and publicly available on Kaggle[6], comprises posts obtained from the "SuicideWatch" subreddits on the Reddit platform. These posts were collected using the Pushshift API, encompassing posts made between December 16, 2008, and January 2, 2021. The dataset consists of 237,074 data points and is divided into two categories: suicide and non-suicide, with an equal distribution of 116,037 data points in each category. On average, the length of each data point in the dataset is approximately 700 words. However, due to computational hardware constraints, in this paper our study focuses on a subset of 120,000 randomly selected data points, with an equal distribution of 60,000 data points for both the suicide and non-suicide categories. The newly constructed dataset does not contain any null values in the text or class values.

III. THE MODELS

In this section, we detail the working of model architectures based on Long Short Term Memory (LSTM) model [7] and the DistilBERT model [8], where these models were calibrated to the social media dataset separately.

A. Long Short Term Memory Model

LSTM: Long Short-Term Memory (LSTM) is a specific type of Recurrent Neural Network (RNN) architecture developed to address the challenge of vanishing or exploding gradients encountered in traditional RNNs. LSTMs, in particular, are highly effective in processing and predicting sequential data, such as time series, speech, text, handwriting, and classification. They have achieved successful results in various domains, including natural language processing, speech recognition, machine translation, and image

captioning. In our study, we utilize LSTMs for the classification task, specifically for distinguishing between suicidal and non-suicidal content in the given text.
The LSTM model consists of following components:

1. **Cell State (Ct):** The cell state acts as a conveyor belt, allowing information to flow through the entire chain of LSTM units. It helps the LSTM remember or forget information over long sequences.
2. **Input Gate (i):** The input gate determines how much of the new input should be added to the cell state. It takes into account the previous hidden state (ht-1) and the current input (xt), and produces a value between 0 and 1 for each element of the cell state.
3. **Forget Gate (f):** The forget gate decides which information to discard from the cell state. It takes the previous hidden state (ht-1) and the current input (xt) as inputs, and produces a forget factor between 0 and 1 for each element of the cell state.
4. **Output Gate (o):** The output gate determines the output of the LSTM unit. It considers the previous hidden state (ht-1) and the current input (xt), and produces an output between 0 and 1.
5. **Hidden State (h):** The hidden state is the output of the LSTM unit. It is a filtered version of the cell state, controlled by the output gate. The hidden state captures relevant information from the input sequence and carries it forward to the next time step.

During the forward pass, the LSTM takes a sequence of inputs (x1, x2, ..., xn) and processes them one by one, updating its internal states at each time step. The input and forget gates regulate the flow of information into and out of the cell state, while the output gate controls the amount of information that is passed to the next hidden state.

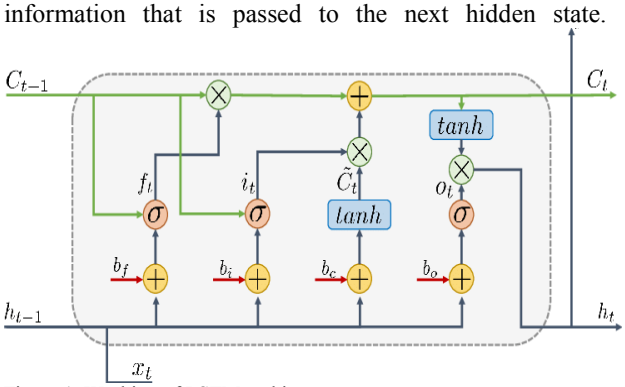

Figure 1. Working of LSTM architecture

As shown in table 1, for our model architecture we use 2 LSTMs layers each of have 100 LSTM Units or cells and an Embedding layer and one dense layer for classification using softmax as activation function and we choose the categorical cross-entropy as loss function and adam as optimizer we evaluate our model on accuracy metrics and add dropout layers after every LSTMs layers whose dropout rate is 0.2 and we kept the learning rate default and choose the batch size of 128 and length of each sentence equal to 200.

Table 1. Hyperparameter values for LSTM model

S.NO.	Hyper Parameter	value
1.	LSTM Units	100
2.	Dropout Rate	0.2
3.	Learning Rate	0.001
4.	Batch Size	128
5.	Sequence Length	200
6.	Number of Epochs	17(30 early stopping)

B. DistilBERT Model

DistilBERT: DistilBERT is a variant of the popular BERT (Bidirectional Encoder Representations from Transformers) architecture. DistilBERT aims to reduce the computational resources required by BERT while maintaining similar performance.

The working of DistilBERT involves following key components:

Transformer Encoder: DistilBERT is built upon the transformer encoder architecture. Transformers consist of self-attention mechanisms and feed-forward neural networks, allowing them to capture contextual relationships and dependencies within a sequence of tokens effectively.
Distillation: The main technique used in DistilBERT is knowledge distillation. It involves training a smaller and faster model (DistilBERT) to replicate the behavior and knowledge of a larger and more computationally expensive model (BERT). The smaller model learns from the predictions and hidden representations of the larger model, allowing it to benefit from its knowledge.
Parameter Sharing: DistilBERT reduces the number of parameters compared to BERT by employing parameter sharing. Specifically, it shares the parameters between the layers of the transformer encoder, which significantly reduces the model's size and computational requirements.
Task-Specific Layers: DistilBERT adds task-specific layers on top of the transformer encoder. These layers are used to fine-tune the model for specific natural language processing (NLP) tasks, such as sentiment analysis, named entity

recognition, question answering, and text classification. These task-specific layers are trained along with the pre-trained transformer encoder during the fine-tuning process.

During the pre-training phase, DistilBERT follows a similar approach to BERT. It is initially trained on a large corpus of unlabeled text in a self-supervised manner, where it learns to predict masked words within sentences and perform next sentence prediction tasks. In the fine-tuning phase, DistilBERT is further trained on task-specific labeled datasets. The pre-trained transformer encoder is frozen, and the task-specific layers are trained using labeled data from the specific task. The model is optimized using gradient-based optimization techniques such as stochastic gradient descent (SGD) or Adam.

DistilBERT achieves a balance between computational efficiency and model performance. Although it sacrifices some model capacity compared to BERT, it still retains much of the knowledge and capabilities of the larger model. This makes DistilBERT well-suited for scenarios where computational resources are limited or where fast inference is crucial, while still achieving competitive performance on various NLP tasks.

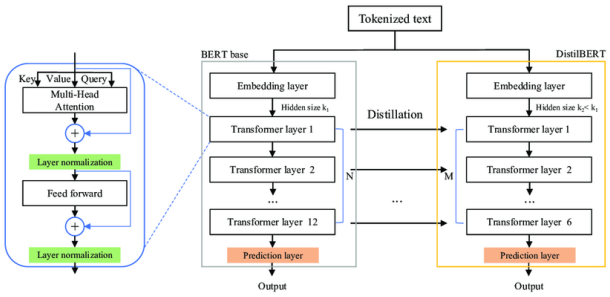

Figure 2. Working of DistilBERT architecture

In Table 2, we present the DistilBERT model with the learning rate set to 5e-5. The training process operates for one epoch, utilizing a batch size of 32 for training and 20 for evaluation. We implemented a warm up period of 500 steps, and logging will take place every 500 steps as well. We set the weight decay for the model at 0.01. The evaluation strategy is determined based on steps.

Table2. Hyperparameter values for DistilBERT model.

S.No.	Hyper- Parameter	Values
1.	Learning Rate	5e-5
2.	Number of Epochs	1
3.	Training Batch Size	32
4.	Evaluation Batch Size	20
5.	Warmup Steps	500
6.	Logging Steps	500
7.	Weight Decay	0.01
8.	Evaluation Strategy	Steps

IV. METHOD

A. EVALUATION METRICS

Evaluation metrics used to determine the effectiveness of the proposed methods can be found underneath:

- True Positive (TP): an outcome where the model correctly predicts the positive class.
- True Negative (TN): an outcome where the model correctly predicts the negative class.
- False Positive (FP): an outcome where the model incorrectly predicts the positive class.
- False Negative (FN): an outcome where the model incorrectly predicts the negative class.

1. Accuracy: It measures the overall correctness of the model's predictions by comparing the number of correct predictions with the total number of predictions. We calculate the accuracy using the following equation:

$$\text{Accuracy} = (TN + TP) / (TN + TP + FP + FN)$$

2. Precision: Precision calculates the proportion of correctly predicted positive instances out of all instances predicted as positive. It is useful when the focus is on minimizing false positives. We calculate the precision using the following equation

$$\text{Precision} = (TP) / (TP + FP)$$

3. Recall: Recall calculates the proportion of correctly predicted positive instances out of all actual positive instances. It is useful when the goal is to minimize false negatives. We calculate the Recall using the following equation:

$$\text{Recall} = (TP) / (TP + FN)$$

4. F1 Score: F1 Score is the harmonic mean of precision and recall. It provides a single metric that balances both precision and recall. We calculate the F1 Score using the following equation:

F1 = (2 * Precision * Recall) / (Precision + Recall)

V. RESULTS

The training and validation accuracy graph in figure 3 demonstrated a consistent upward trend, the validation accuracy closely followed the training accuracy, suggesting minimal overfitting and generalization of the model.

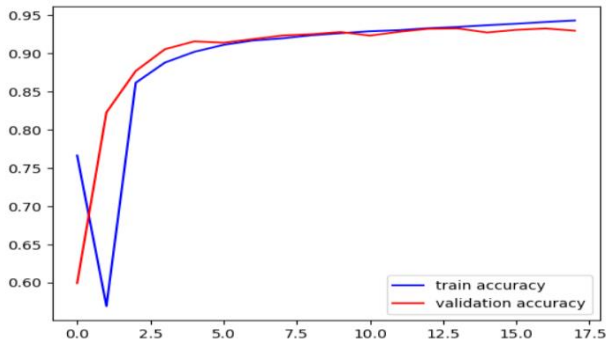

Figure 3: Graph Training accuracy Vs. Validation accuracy of LSTM

The training and validation loss graph exhibited in figure 4 shows a steady decrease over epochs, indicating effective model optimization. The validation loss closely tracked the training loss, indicating good generalization and minimal overfitting.

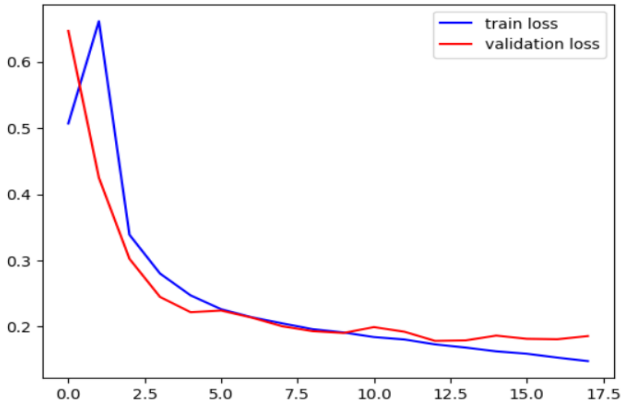

Figure 4: Graph Training Loss Vs Validation Loss of LSTM

The fine-tuning process resulted in a notable increase in accuracy, as shown in the graph of figure 5. The accuracy steadily improved over steps, demonstrating the effectiveness of fine-tuning for enhancing the model's performance. The results highlight the successful adaptation of the pre-trained model to suicidal text detection.

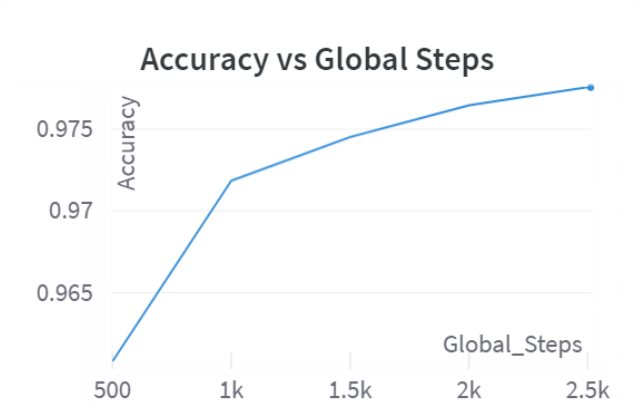

Figure 5: Evaluation accuracy of DistilBERT architecture

The fine-tuning process yielded a notable decrease in loss, as depicted in figure 6. The loss consistently decreased over iterations, indicating successful fine-tuning and improved model convergence. These findings demonstrate the effectiveness of fine-tuning for refining the model's performance and optimizing the task-specific for suicidal text detection.

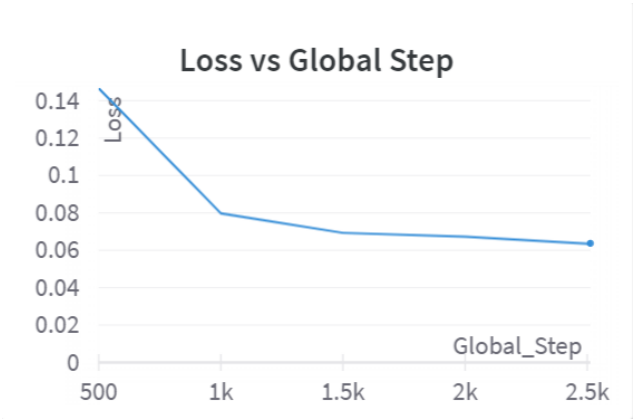

Figure 6: Evaluation Loss of DistilBERT architecture

As shown in table 3, the LSTM model achieved an accuracy of 0.9326 and a corresponding loss of 0.1793 with a sequence length of 200 words. However, the DistilBERT model with a sequence length of 512 words, showed a significant improvement in accuracy to 0.9776, along with a much lower loss of 0.0637. These results highlight the superior performance of DistilBERT compared to LSTM, indicating its effectiveness in effectively handling longer sequences of text and hence increasing the chances of accurate detection of suicidal text.

Table 3. Accuracy and Loss with respect to architecture and sequence length

S.No.	Architecture	Sequence Length(words)	Accuracy	Loss
1.	LSTM	200	0.9326	0.1793
2.	DistilBERT	512	0.9776	0.0637

The confusion metrics for each of the LSTM and DistilBERT model architectures that have been considered in this paper are shown in Table 4. In the case of the LSTM model, when non-suicide instances were classified, a precision of 0.95, recall of 0.91, and an F1-score of 0.93 were achieved, with a support of 18050 instances. Similarly, when suicide instances were classified, a precision of 0.91, recall of 0.95, and an F1-score of 0.93 were attained, with a support of 17950 instances.

In contrast, the DistilBERT model classified non-suicide instances with, a precision of 0.98, recall of 0.98, and an F1-score of 0.98, with a support of 19877 instances. Similarly, when suicide instances were classified, a precision of 0.98, recall of 0.98, and an F1-score of 0.98 were achieved, with a support of 19723 instances. The DistilBERT architecture therefore, accurately classified both non-suicide and suicide instances as is highlighted by these results. Also, the result clearly showed that the DistilBERT model that took the longer sequence length performed better than LSTM which took a shorter sequence length.

Table 4. Confusion metrics with respect to architecture and each class

Architecture	Sequence Length(words)	Precision	Recall	F1-score	Support
LSTM	non-suicide	0.95	0.91	0.93	18050
	suicide	0.91	0.95	0.93	17950
DistilBERT	non-suicide	0.98	0.98	0.98	19877
	suicide	0.98	0.98	0.98	19723

VI. DISCUSSION & CONCLUSIONS

Based on the accuracy and loss data presented in the table and graphs, we can conclude that DistilBERT outperforms other architectures. DistilBERT is particularly well-suited for general domains, as it effectively handles large sequence lengths and readily accommodates various words that help classify suicidal content and ideation.

In this research paper, two models are proposed for the analysis and detection of suicidal content across various social media platforms. Currently, our focus has been on deploying the model to Telegram using the Telegram Bot API. The deployed model reads messages sent by users and responds accordingly if any suicidal content is detected. Responses may include sending motivational quotes or alert messages to the user's close friends or family members, providing support and intervention for suicidal ideation. To train our models, we utilized data from the "SuicideWatch" subreddit on Reddit [6]. We trained LSTM and performed fine-tuning of the DistilBERT architecture. LSTM performs best for sequence lengths below 200, but the model's performance significantly declines for longer sequences. DistilBERT is trained on sequences longer than 512 using the parameter truncation set to true.

The main contribution of this work lies in enhancing the quality of suicidal content detection through fine-tuning the DistilBERT model. Moving forward, there is potential for using larger datasets to achieve better fine-tuning or incorporating more advanced architectures such as BERT or ChatGPT. Additionally, expanding the data sources beyond Reddit to scrape from various platforms and deploying the model for real-time detection across multiple social media platforms, rather than just Telegram, would greatly benefit individuals in obtaining appropriate assistance to address suicidal ideation Some of these ideas and others form the immediate next steps for us to pursue in the near future.

ACKNOWLEDGMENTS

The authors are grateful to Indian Institute of Information Technology, Una for providing computational resources for this paper.

REFERENCES

[1] Kochanek, J. Q. Xu, and E. Arias, "Mortality in the United States, 2019," NCHS Data Brief, no. 395, Hyattsville, MD: National Center for Health Statistics, 2020.

[2] G. Astoveza, R. J. P. Obias, R. J. L. Palcon, R. L. Rodriguez, B. S. Fabito, and M. V. Octaviano, "Suicidal behavior detection on Twitter using neural network," in TENCON 2018-2018 IEEE Region 10 Conference, 2018, pp. 0657-0662.

[3] M. K. Nock, G. Borges, E. J. Bromet, C. B. Cha, R. C. Kessler, and S. Lee, "Suicide and Suicidal behavior," Epidemiologic Reviews, vol. 30, no. 1, pp. 133-154, 2008.

[4] M. M. Tadesse et al., "Detection of suicide ideation in social media forums using deep learning," Algorithms, vol. 13, no. 1, 2019.

[5] T. H. H. Aldhyani et al., "Detecting and analyzing suicidal ideation on social media using deep learning and machine

learning models," International Journal of Enviormental Research and Public Health, vol. 19, no. 19, pp. 12635,2022.

[6] Nikhileswar Komati, "Suicide and Depression Detection," Kaggle 2021.[Online]. Available: https://www.kaggle.com/datasets/nikhileswarkomati/suicide-watch [Accessed: 10 Feb 2023].

[7] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

[8] Sanh, Victor, et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." arXiv preprint arXiv: 1910.01108 (2019).