Endsem Report for Suicide Ideation Prediction from Social Media Conversations

Medha Hira
IIITD
medha21265@iiitd.ac.in

Arnav Goel
IIITD
arnav21519@iiitd.ac.in

Siddharth Rajput
IIITD
siddharth21102@iiitd.ac.in

Abstract

Amid modern life's complexities, the convergence of personal and professional demands reveals a stark reality: a concerning rise in mental health issues. This paper underscores the pressing need to detect and address suicide ideation due to the alarming rise in suicide rates. From 2000 to 2021, there has been a 36% increase in suicide rates, emphasizing the urgency of effective prevention and intervention strategies (Source). Suicide causes profound emotional and psychological impact on individuals, families, and communities. Recognizing suicidal thoughts at their nascent stages opens the gateway to timely intervention, offering the potential to thwart individuals from progressing to the dire precipice of suicide attempts. We also deploy our model in the real world by preparing a Reddit Bot to detect if a post has a suicidality ideation risk when tagged.

With this backdrop in mind, our primary objective is to engineer a robust predictive model capable of identifying individuals at heightened risk of Suicide, particularly within the context of their online activities. The reason for this approach stems from the alarming prevalence of youth seeking solace in the virtual realms of social media platforms, often voicing their innermost thoughts under the deceptive cloak of anonymity.

1. Introduction

As we spend most of our time plugged into our devices, we aim to prepare a system to detect suicide ideation through social media activity. Many users feel more comfortable expressing their feelings and thoughts online, often under the cloak of anonymity. This can lead to more honest and unfiltered discussions about sensitive topics like mental health and Suicide.

As asserted in [1], it is increasingly evident that individuals experiencing suicidal ideation frequently share their thoughts and emotions on social media platforms. Consequently, there is an opportunity to validate the presence of suicide ideation by analyzing these social media posts. Detecting signs of suicidal ideation early on can play a pivotal role in providing timely access to assistance for individuals in need.

We aim to develop a scalable machine learning system to detect signs of suicide ideation within Reddit posts. By doing so, we hope to bridge the gap between the digital world and mental health support networks. Detecting these risk factors early can facilitate quicker access to help for those suffering silently and contribute to destigmatizing mental health issues. Our motivation is to democratize access to assistance and revolutionize how we address this sensitive issue in the digital age, enhancing the timeliness and effectiveness of our responses.

To build this system, we are utilizing a publicly available text dataset from the r/SuicideWatch subreddit. We have implemented a thorough data preprocessing phase and fine-tuned hyperparameters to optimize the performance of our machine-learning models. These models encompass a variety of approaches, such as Principal Component Analysis (PCA) for dimensionality reduction and a selection of different algorithms, including Logistic Regression, Support Vector Machines (SVM), Naive Bayes, Decision Tree, and Random Forest. In our evaluation process, we not only measure the accuracy of these models but also assess their latency on the same computing device. It lets us weigh the trade-offs between model responsiveness and accuracy, helping us identify the most suitable model for our specific application.

2. Literature Survey

Suicide represents a critical issue that demands our attention. To save lives, it is essential to detect and prevent suicide attempts at an early stage. [6] presented a pioneering survey that delves into suicidal ideation detection methods. It categorizes these methods into two primary domains: clinical approaches involving the interaction between professionals and individuals in distress and machine learning techniques that harness feature engineering or deep learning for automatic detection based on online social content. The paper further examines domain-specific applications of these methods, drawing from various data sources such as

questionnaires, electronic health records, suicide notes, and online user-generated content.

[7] examines the relationship between suicidal ideation and subsequent Suicide using various statistical measures. It finds a moderate association. It highlights that individuals' expression of suicidal ideation is indicative of their psychological distress.

[4] performed a comparative analysis on suicidal ideation detection using NLP, Machine, and Deep Learning. They evaluated the performance of DL classifiers (Long-Short Term Memory (LSTM), Bi-directional LSTM (BiLSTM), Gated Recurrent Unit (GRU), Bi-directional GRU (BiGRU), and combined model of CNN and LSTM (C-LSTM)) with traditional ML approaches (Random Forrest (RF), Support Vector classifier (SVC), Stochastic Gradient Descent classifier (SGD), Logistic Regression (LR), and Multinomial Naive Bayes (MNB) classifier).

[5, 11] use hybrid deep learning and machine learning for the automated detection of suicidal ideation from social media, achieving high accuracy through text analysis and feature-based classification.

[2] highlights that the commonly used methods for feature extraction include LIWC, LDA, LSA, and Word2vec, with NMF and PCA used less frequently. Support Vector Machine (SVM) is the most commonly used technique (62.5%), followed by Decision Trees, Logistic Regression, and Random Forest. It emphasizes the need for future work classifying posts, estimating the risk of Suicide, and analyzing and optimizing predictive parameters. Further, it encourages analyzing the temporal component of user posts for sentiment.

3. DataSet

The University of Maryland Reddit Suicidality Dataset is a dataset sourced from Reddit, an online platform where users converse anonymously on diverse topics. This dataset was first introduced in [12]. It supports research in suicide risk assessment and prevention.

This dataset mentioned above was constructed from the Full Reddit Submission Corpus [Link]. Posts were picked up from the "r/SuicideWatch" subreddit for the years 2006 to 2015. To assess the level of suicide risk, users in the dataset are categorized into four categories: no risk, low risk, moderate risk, and severe risk. These annotations were carried out through a combination of expert assessments and crowd-sourcing.

For further exploration of this dataset and to emphasize the importance of suicide risk detection and prevention, a shared task was launched in the 2019 Workshop on Computational Linguistics and Clinical Psychology [13].

This dataset mentioned above requires mailing the authors, and we plan to acquire it for the next leg of our project. We, however, find a similar dataset on Kaggle, i.e. Suicide and Depression Detection. This dataset has been curated by collecting posts from "r/SuicideWatch" from 2009 to 2021. It contains 287000 posts classified into two classes, i.e. **Suicide** and **Non-Suicide**. Both classes have an equal number of samples. The classwise distribution of the dataset is equal. This indicates that the data is **not skewed**.

3.1. Text Pre-processing

As the data contains social media data, it contains a lot of noise in the form of non-ASCII characters, URLs, and social media handles. We thus perform the following standard text-preprocessing on our text data:

1. Removal of non-ASCII characters like emojis
2. Removal of URLs
3. Removal of Usernames and Social Media Handles
4. Punctuation Removal
5. Removal of Stopwords: These are commonly occurring insignificant words in a language which do not hold importance for predicting target variables.
6. Lowercasing: All the text is lower-cased to ensure the same meaning of the word to be delivered across

3.2. Exploratory Data Analysis

Word Frequency Analysis: 3.2 shows the frequency of the most commonly occurring words in the entire dataset after preprocessing. This shows that the words are general, and simply looking will not help us predict the class of the data.

K-Means Clustering: Silhouette scores were computed for a range of clusters (k=2 to 11) 3.2 using the k-means algorithm. Notably, the silhouette scores reached their maximum at k=3, indicating optimal cluster separation. Subsequently, k-means clustering was applied to the TF-IDF embedded data with k=3, yielding insightful patterns in the dataset. 3.2

Post Length Analysis: This shows the longest and shortest posts' length, highlighting the dataset's diversity.

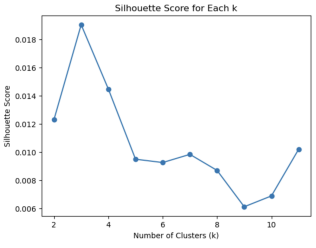

Figure 1. Silhouette Score

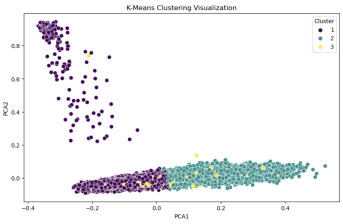

Figure 2. Clusters for k = 3

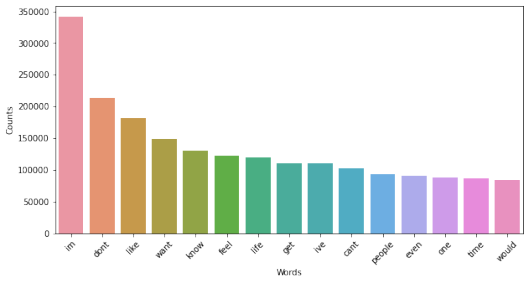

Figure 3. Word Frequency Analysis

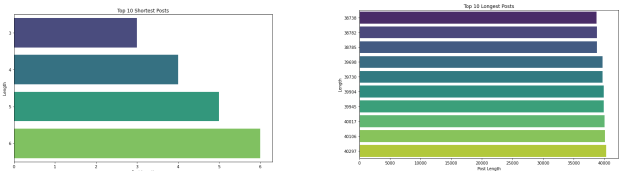

Figure 4. Post Length Analysis

Word Cloud: This shows the word clouds for both classes respectively. These word clouds show that the words in both classes are generic. They are words humans would associate with death, Suicide and trauma. Thus, it is not possible to predict suicidal ideation by simply looking at words in the tweet.

4. Methodology

Since the data consists of around 287000 posts, we randomly sample 15000 posts from each class to curate a smaller custom dataset of size 30000 with equal distribution of the classes. We split this into training and testing data with a 75:25 split amounting to 22500 training samples and 7500 testing samples.

4.1. Word Embeddings and Vectorisation

Converting text data into features is crucial for applying machine learning models, and this process has garnered significant recognition in utilizing machine learning techniques on textual data. Two prominent methods emerge in this context: **Vectorization** and **Word Embeddings**. To accomplish this task, we utilize the following techniques:

1. **Tf-Idf** (Term Frequency-Inverse Document Frequency) is a statistical method to evaluate the importance of a word in a document relative to a collection of documents compared to its frequency. We set the embedding size as 1000 here.

2. **Word2Vec** [8] converts text data into word embeddings by capturing the semantic relationships and the context. We set the size of the embedding as 512 here.

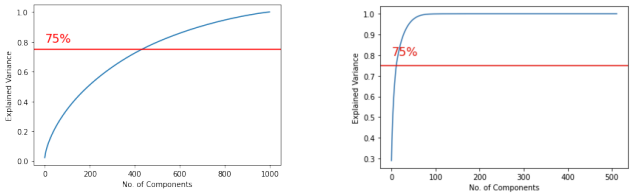

Figure 5. Comparison of PCA results between TF-IDF (left) and Word2Vec (right)

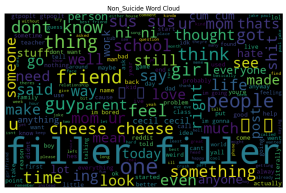

Figure 6. Word Cloud for Non-Suicide Labelled Posts

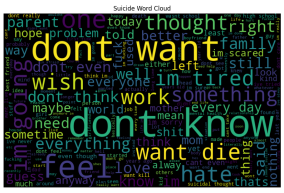

Figure 7. Word Cloud for Suicide Labelled Posts

3. **GLoVe** [9] creates dense word embeddings using the global occurrence of words and creates vectors. We also use Transformer-based word embeddings i.e. **BERT** (bert-base-uncased) [3] and **Sentence-BERT** [10]. Both are pre-trained models, with the latter being very fast.

DL models beyond ANNs are not allowed, so we only used TF-IDF embeddings of size 1000 and Word2Vec Embeddings of size 512 for running machine-learning models. The rest of the embeddings are created and dumped. Principal Component Analysis was performed on both embeddings to preserve 75% of the variance. Figure 5 shows that **434 components** preserve 75% in TF-IDF embeddings while it only requires **13 components out of 1000** to preserve 75% variance in Word2Vec Embeddings as shown by Figure 6.

The figure below shows the t-SNE visualization of the TF-IDF embeddings and Word2Vec embeddings. It reduces it to 2-dimensions and helps us visualize the separability between the two classes:

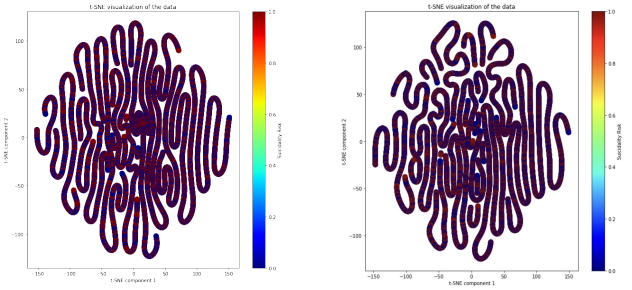

Figure 8. TSNE: TF-IDF(left), Word2Vec(right)

4.2. Evaluation Metrics

Evaluation metrics are a quantitative analysis to assess the effectiveness of a machine learning model. Below are the metrics that we've reported:

- **Accuracy**: Measures the overall correctness of the model. It is the ratio of correct predictions to the total number of predictions or tests.
- **Precision**: Assesses true positives among all positives.
- **Recall**: Measure of our model correctly identifying true positives.

4.3. Models Used

Ours is a **binary classification task** as we aim to predict if a given piece of text (social media post) should be classified as Suicide (has suicide ideation) or non-suicide (free of risk). We ran various Machine Learning Models and Ensemble Methods. We additionally trained an MLP classifier, i.e. an ANN, on our embeddings. Grid search was run along with **5-fold cross validation** to find the best hyperparameters for each model.

A summary of the models and the hyperparameters found:

- For **Logistic Regression**, the L2 penalty was applied for regularisation with $\lambda = 1$. The solver chosen was "newton-cg", with a maximum of 1000 iterations.
- For the **Decision Tree Classifier**, the "Gini" criterion was found to be the best. The maximum depth of the tree was restricted to 10. The minimum number of samples to be split was set to 5, and the minimum number of leaves was set to 1.
- For the **Random Forest Classifier**, a maximum depth of 20 was specified, and 200 decision-tree estimators were taken for making predictions.
- For the **k-Nearest Neighbors Classifier**, on cross-validation $k = 3$ was found. The distance metric was chosen to be Euclidean.
- **Linear Discriminant Analysis** increases separability between the classes and makes them tighter. It does not have a lot of hyperparameters to fine-tune and reduces the dimensionality of the data to $C - 1$, where C refers to the number of classes.
- Two variants of the **Naive Bayes** model were run, i.e. Gaussian NB and Bernoulli NB. Both do not have hyperparameters to tune.
- **Support Vector Machine Classifier** was employed. The SVM Classifier was trained with three kernels: i.e. Linear, Polynomial and RBF. Regularisation was applied to reduce the chance of overfitting.
- **Ensemble Methods** were employed such as **AdaBoost Classifier**, **XGBoost Classifier**, **Stacking Classifier** and **Voting Classifier**. The Stacking Classifier is a 2-level architecture comprising a Random Forest Classifier stacked with a Logistic Regression model on the top. The Voting Classifier takes a Hard Vote between three base classifiers.
- We also trained a **MultiLayer Perceptron** on our dataset. First, we tried one with two hidden layers of size (128,64) and another one with three hidden layers of size (128,64,32). We tried running our model with the four activation functions, i.e. 'relu', 'logistic', 'identity' and 'tanh'. A batch size of 128 was set to speed up the training process.

While running our experiments, we noted the time it took for the model to train. In Section 5, we will conduct a comprehensive analysis to assess the intricate trade-offs between various evaluation metrics and the crucial aspect of latency. In a real-world scenario, achieving precise predictions is paramount, but minimizing computational expenses is equally important, making the model easier to deploy and user-friendly.

5. Results & Analysis

Figure 9 and 10 shows the models we ran and the values of various evaluation metrics we mentioned in section 4.2. As reported, we trained the respective models on TF-IDF and Word2Vec embeddings generated for the preprocessed Reddit posts. We reported the best scores of either of the two embeddings for each model.

Analysis of Machine Learning Models: The best-performing models are LDA, Logistic Regression and the SVM classifier. SVM showed the best accuracy with a **Polynomial** kernel, and we report its results. However, on closer analysis, we can see that SVM has a broader gap in its training and testing accuracies, indicating **a high variance of the model**. LDA and Logistic Regression offer similar accuracies with lesser chances of the models being overfit.

Using different embeddings also immensely helped in improving model performance. Models like Gaussian and Bernoulli Naive Bayes gave only 55% accuracy with TF-IDF embeddings, while the accuracy increased almost 30 points to 85% accuracy while using Word2Vec Embeddings. A similar trend was observed for k-nearest Neighbors and Random Forest Classifiers. This is primarily due to TF-IDF embeddings merely being sort of a frequency-based mathematical metric which are not able to exploit geometrical properties associated with embeddings. Word2Vec embeddings, on the other hand, are prepared using Neural Networks, which help project texts into a fixed-dimensional

Model	Embedding	Accuracy		Precision		Recall	
		Train	Test	Train	Test	Train	Test
Logistic Regression	TF-IDF	92.11	91.50	92.00	92.00	92.00	92.00
	TF-IDF-PCA	91.44	91.34	91.00	91.00	91.00	91.00
Gaussian Naive Bayes Classifier	Word2Vec	85.73	86.01	86.00	86.00	86.00	86.00
	Word2Vec -PCA	85.73	86.01	86.00	86.00	86.00	86.00
Bernoulli Naive Bayes Classifier	Word2Vec	86.21	86.45	87.00	87.00	86.00	86.00
	Word2Vec -PCA	83.87	84.25	84.00	84.00	84.00	84.00
Support Vector Machine Classifier	TF-IDF	97.05	91.93	97.00	92.00	97.00	92.00
	TF-IDF-PCA	96.57	91.54	97.00	92.00	97.00	92.00
Decision Tree Classifier	TF-IDF	88.21	85.36	89.00	86.00	88.00	85.00
	TF-IDF-PCA	93.79	84.66	94.00	85.00	94.00	85.00
Linear Discriminant Analysis	TF-IDF	92.06	91.30	92.00	91.00	92.00	91.00
	TF-IDF-PCA	91.22	90.93	91.00	91.00	91.00	91.00
k-Nearest Neighbors Classifier	Word2Vec	90.85	86.26	91.00	88.00	91.00	86.00
	Word2Vec -PCA	92.15	87.18	92.00	88.00	92.00	87.00

Figure 9. Results of Machine Learning Models

Model	Embedding	Accuracy		Precision		Recall	
		Train	Test	Train	Test	Train	Test
Random Forest Classifier	Word2Vec	95.22	90.48	95.00	91.00	95.00	90.00
	Word2Vec -PCA	91.48	88.69	92.00	89.00	91.00	89.00
AdaBoost	TF-IDF	89.48	89.24	89.00	87.00	89.00	87.00
XGBoost	TF-IDF	85.23	87.45	86.00	85.00	85.00	85.00
Voting Classifier	TF-IDF	93.25	91.42	92.00	91.00	91.00	91.00
Stacking Classifier	TF-IDF	92.74	91.37	91.00	90.00	90.00	89.00
MLP 2-Layer (ReLU)	TF-IDF	93.08	91.50	90.00	90.00	90.00	89.00
MLP 3-Layer (Identity)	TF-IDF	92.93	91.36	88.00	88.00	88.00	88.00
MLP 3-Layer (ReLU)	TF-IDF	93.41	91.45	89.00	87.00	88.00	87.00

Figure 10. Results for Ensemble Methods and MLP

space. Vectors in this space show much better geometrical properties, which greatly help these classifiers (especially Naive Bayes and k-NN).

As presented in our Mid-semester Report, the Random Forest Classifier showed a very high training accuracy and consequent high difference between the training and testing accuracies. This **higher variance** indicates our Random Forest Classifier was being overfitted. We set a maximum depth and pruned our trees to reduce this overfitting. We saw a substantial rise in the testing accuracy and a reduction in the gap.

Analysis of the Ensemble Methods: Ensemble models are based on the concept of combining multiple models (weak learners), each with high diversity. We employed five ensemble techniques, i.e. Random Forest Classifier, AdaBoost, XGBoost, Voting and Stacking Classifier. Voting and Stacking Classifiers gave the highest accuracies, and that comes intuitively as they correct the mistakes of weaker classifiers. An interesting anomaly is observed wherein our test accuracy using the XGBoost Classifier is higher than its training accuracy. The Random Forest classifier is explained in the previous paragraph.

Analysis of the MLP Classifier: We used this deep learning technique and trained our dataset using two hidden layers of size (128,64) and then three hidden layers of size (128,64,32) MLP. We got an accuracy of 91.50% using two-layer MLP with ReLU activation function and 91.45% using three-layer MLP with ReLU. The identity function gave a test accuracy of only 91.36%.

Figures 11-14 plot the Validation Loss vs Epochs and Training Loss vs Epochs curves for our three-layer MLP for all the four activation functions. As we see ReLU's validation loss is increasing. This is due to the problem of exploding gradients. Mentioned here are some hyperparameters chosen:

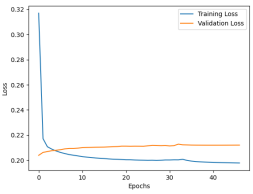

Figure 11. MLP Identity

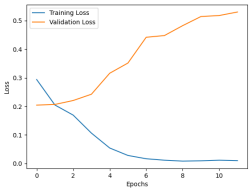

Figure 12. MLP ReLU

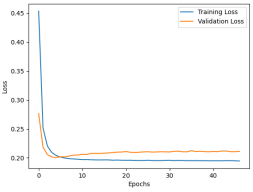

Figure 13. MLP Tanh

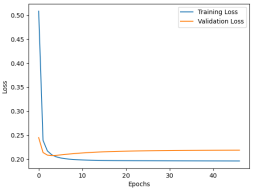

Figure 14. MLP Logistic

2 layer MLP (ReLU): 'batch size': 32, 'learning rate init': 0.001
3 layer MLP (ReLU): 'batch size': 64, 'learning rate init': 0.001
3 layer MLP (Identity): 'batch size': 128, 'learning rate init': 0.1

6. Model Deployment - Reddit Bot

We created a Reddit bot utilizing PRAW and our project's most effective machine learning model, focused on predicting suicidal ideation on the platform. The bot preprocesses post content with natural language processing and assesses it using the top-performing ML model. If a risk is detected, it provides mental health resources and supportive messages, designating the post as safe for non-concerning content. This strategy leverages our best ML model for precise risk prediction, aiming to offer timely support to Reddit users.

7. Conclusion and Learnings

Thus, we conclude our project wherein we tried various ML, Ensemble and Neural Network-based models to aid the process of Suicide Ideation Detection from Social Media Conversations. We compared various models and better understood the bias-variance trade-off to understand performance on Out-of-Distribution Data better. We also learnt about the deployment of ML models on social media websites by developing a Reddit Bot which can detect suicide ideation from Reddit Posts. We greatly value this experience and hope to improve upon our work by trying more advanced architectures and tougher datasets.

8. Individual Member Contributions

All the members contributed equally to the work and helped each other edit the codes and write this report. The individual contributions listed below are only representations of the assignments of tasks to each team member.

- **Arnav:**
  Ensemble Methods, Word2Vec Embeddings, Text preprocessing, EDA, Logistic Regression, Naive Bayes, Linear Discriminant Analysis, k-NN classifier, Report Writing Principal Component Analysis.

- **Medha:**
  MLP Classifier, Ensemble Methods, TF-IDF embeddings, GLoVe Embeddings, EDA, Decision Tree Classifier, Random Forest Classifier, Report Writing, Presentation

- **Siddharth:**
  Reddit Bot (Model Deployment), Text preprocessing, Transformer and Word2Vec Embeddings, Principal Component Analysis, Report Writing and Presentation

References

[1] Thamer H H Aldhyani, Saud N Alsubari, Abdulrahman S Alshebami, Hussain Alkahtani, and Zeeshan A T Ahmed. Detecting and analyzing suicidal ideation on social media using deep learning and machine learning models. *International Journal of Environmental Research and Public Health*, 19(19):12635, Oct. 2022.

[2] G Castillo-Sánchez, G Marques, E Dorronzoro, and et al. Suicide risk assessment using machine learning and social networks: a scoping review. *J Med Syst*, 44:205, 2020.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[4] R Haque, N Islam, M Islam, and MM Ahsan. A comparative analysis on suicidal ideation detection using nlp, machine, and deep learning. *Technologies*, 10(3):57, 2022.

[5] Pratyaksh Jain et al. Depression and suicide analysis using machine learning and nlp. *Journal of Physics: Conference Series*, 2161:012034, 2022.

[6] Shaoxiong Ji, Shirui Pan, Xue Li, Erik Cambria, Guodong Long, and Zi Huang. Suicidal ideation detection: A review of machine learning methods and applications. *IEEE Transactions on Computational Social Systems*, 8(1):214–226, 2021.

[7] C. M. McHugh, A. Corderoy, C. J. Ryan, I. B. Hickie, and M. M. Large. Association between suicidal ideation and suicide: meta-analyses of odds ratios, sensitivity, specificity and positive predictive value. *BJPsych Open*, 5(2):e18, 2019. Erratum in: BJPsych Open. 2019 Mar;5(2):e24.

[8] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[9] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.

[10] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[11] Amit Roy, Kim Nikolitch, Ryan McGinn, et al. A machine learning approach predicts future risk to suicidal ideation from social media data. *npj Digital Medicine*, 3:78, 2020.

[12] Han-Chin Shing, Suraj Nair, Ayah Zirikly, Meir Friedenberg, Hal Daumé III, and Philip Resnik. Expert, crowdsourced, and machine assessment of suicide risk via online postings. In *Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic*, pages 25–36, 2018.

[13] Ayah Zirikly, Philip Resnik, Özlem Uzuner, and Kristy Hollingshead. CLPsych 2019 shared task: Predicting the degree of suicide risk in Reddit posts. In *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology*, pages 24–33, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.