Ensemble NLP for Mental Health Warning Sign Detection

Param Damle psd9vgc@virginia.edu
Richard Wang rxw2cxy@virginia.edu
Kabir Menghrajani km5qte@virginia.edu

1 Introduction

Mental health is a serious issue that affects millions of people each year. According to the National Institute of Mental Health (NIMH), more than one in five adults in the US have a mental illness (57.8 million in 2021). It is even more prevalent in young adults aged 18-25 with a rate of 33.7%, who are the primary demographic who post and comment on social media. Our project uses multiple datasets to try and classify messages which may be indicative of several types of mental illnesses. On top of this, it has the purpose of being able to identify which words in the message in question were the major signals that something could be wrong. Our hope is that we can use NLP to try and identify early if someone may have a mental illness so that they can get help before it worsens; the main purpose of this system would be a "Red Flag Detector" that would alert mental health providers to members of their community who may need treatment.

2 Prior Work

Loke Pak-Yen found several existing open source pre-trained models for mental health prediction using NLP such as Bidirectional Encoder Representations from Transformer (BERT) and compared the results of running each on some text such as the results on Google after searching "depression". One of the main takeaways to consider was that there are several biases and ethical concerns to be aware of when using NLP for mental health related applications. These biases mainly involve using biased data which may not be representative of the population as a whole. This means the need for quality data is extremely high otherwise models may end up perpetuating biases in the data, which we kept in mind.

Chua et al. (2022) developed two models for the use of NLP with mental health. The first one was a lightweight feature based model which used word grams. The other model was a multi-task deep neural network which consisted of several task-specific layers. They found that the lightweight model performed well across almost all domains and tasks. They also mention the importance of ethical concerns and biases when it comes to using these language models, and cite an example where there are no gender labels for the participants in the data which may result in an uneven distribution.

3 Data

3.1 Anxiety and Depression

We used the Anxiety and Depression Dataset to train our model to learn signs of anxiety and depression in text. It contains 6896 entries of text labeled as 1 to indicate anxiety/depression and 0 to indicate no anxiety/depression. There are 733 entries labeled as 1, and 6247 labeled as 0.

3.2 Human Stress

We used the Human Stress Dataset to train our model to predict stress in text samples. The dataset contains 2343 entries sourced from various mental health subreddits, and is labeled as 0 for no stress and 1 for indicates stress. 21% of the entries come from the r/ptsd subreddit and 19% come from the r/relationships subreddit.

3.3 Student Depression

We used the Student Depression Dataset to train our model to identify text that indicates depression. This dataset is comprised of 7489 entries sourced from various social media platforms, with posts following English grammar from 15-17 yeara old students. The five columns of this dataset are text, labels, age, age category, and gender.

3.4 Suicidal Tweets

We used the Suicidal Tweets Dataset to train our model to identify text that is indicative of suicidal thoughts or tendencies. This imbalanced dataset contains 1778 tweets, with about 37% being potential suicide posts and about 63% being non-suicide posts.

4 Methods

4.1 Classification Techniques

Across all our datasets, we want to apply uniform tokenization and produce a word embedding vector for each token in a string, combining to form an embedding vector for each text. This set of vectors will be split into separate training and validation sets for each dataset; our methods used a 75/15 split. Thus, by the end of pre-processing, each of our 4 datasets had $|V|$ columns for the $|V|$-dimensional embedding vector for a given training document and a column for the class label (with 1 representing the presence of some red flag).

Our model is comprised of several domain-specific sub-models that each classify based on an individual dataset (anxiety, stress, depression, or suicide). Each model was trained on its dataset for word similarity (using the embedding vectors produced in pre-processing) to texts that are marked with a 1 (class label for exhibiting problematic behavior) and contrasted with training samples marked with a 0; this outputted a class label of 1 or 0 to determine if the input text exhibits the specific problematic behavior or not. Thus, the output for the ensemble model (classifying whether the text presents any red flags as a whole) was a combination of the outputs of the 4 sub-models. Although we initially wanted to logically `OR` the rounded scores, we decided to instead display the full floating point score to allow more nuance in our prediction.

4.2 Evaluation

To measure success, we used a weighted F-β metric as it provides a better measurement of incorrectly classified cases than accuracy alone, and will reduce the effect of unbalanced classes on our evaluation. We will use a $\beta > 1$ weight parameter to assign emphasis to recall as a higher recall will indicate that our algorithm is flagging more of the concerning texts, while a higher precision would indicate that most of the samples flagged are concerning. In the context of predicting issues with mental health, it is more important to identify all possible indicators of concern (to ensure nobody who needs our help slips through) than to ensure the consistency of flagged content. Allowing false positives errs on the side of caution in this case, whereas allowing false negatives does not.

Leveraging cross validation within the training set improved our model generality, to demonstrate that our approach can someday be scaled to other platforms, sources of text, and perhaps even languages.

4.3 Text Highlighting

We sought to further expand a simple class label result by, upon receiving a classification of 1 from the model, scanning through the input text to identify the n-gram, or sequence of words, with the highest similarity to problematic text. Returning the subsection of text most relevant to the mental health issue is a unique expansion of this solution that will support mental health assistance and treatment efforts by highlighting to care providers what specifically triggered the red flag, so that the provider can efficiently make a judgement on whether to follow up or not.

5 Implementation

The code for our project can be found at the following link: LiFESaVeR

5.1 Architecture

Our implementation architecture consisted of three main steps.

1. Our training text is tokenized into one and two grams, with stop words removed.
2. We compute a TF-IDF matrix from each tokenized training corpus.
3. A neural network receives the cosine similarity scores between a sentence in the test set and the sentences in the training set, which utilizes the TF-IDF matrix computed previously. The neural network performs a cross validation at each epoch, and a $F - \beta$ score is calculated to prioritize recall to minimize false negatives. This network outputs a single floating point value in $[0, 1]$ that represents a per-condition red flag score.

The libraries we called for this project included TensorFlow, Keras, Numpy, Matplotlib, and NLTK. We did not use anyone else's code besides these packages.

5.2 Preprocessing

In the preprocessing pipeline, the input undergoes several transformations. Initially, all characters are converted to lowercase for uniformity. Subsequently, stemming is applied to reduce words to their root forms, promoting consistency in semantic representation. The text is then tokenized into both 1-grams and 2-grams, capturing individual words and pairs of adjacent words. Passing both 1- and 2-grams in any order here improves model performance as the TF-IDF method relies merely on frequency and not ordering, so passing n-grams of various lengths together will only provide additional context on the occurrence of token patterns. Stopwords are systematically eliminated, with an exception made for instances where they are combined with a non-stopword in a 2-gram. The provided example illustrates this process, wherein the input sentence, "Here's a random string of words that I put together!" is transformed into a structured output, containing single words and relevant word pairs based on the described tokenization and stopword removal criteria.

Input: `"Here's a random string of words that I put together!"`

Output: `[('random',), ('string',), ('word',), ('put',), ('togeth',), ('a', 'random'), ('random', 'string'), ('string', 'of'), ('of', 'word'), ('word', 'that'), ('i', 'put'), ('put', 'togeth')]`

5.3 TF-IDF

To find the similarity scores between sentences and our training data, we calculated the TF-IDF matrix of our training data, with each condition dataset undergoing its own processing. After the training data was tokenized, each sentence/document was a list of tokens, each of which were a 1- or 2-gram. The first step involved calculating term frequency (TF) vectors for each document, which is a compilation of how many times each token from the entire corpus appears in that document. Afterwards, the inverse document frequency is calculated for each token, which is defined as $\log(\frac{N}{d})$, where N is the total number of documents, and d is the document frequency, the number of documents which the token appears in. We multiply each token TF value within each document vector by the token's IDF to get the TF-IDF vector for each document. This vector is scaled to a magnitude of 1 to allow cosine similarty scoring. Compiling this vector for all documents gives us the final TF-IDF matrix, where the rows are the term dimensions and each column is a document vector. The dimensions are provided below:

Detector	$\|V\|$ (# tokens)	N (# documents)
Anxiety	49608	6980
Stress	86584	2838
Depression	53281	7486
Suicide	22970	1788

5.4 Scoring

For each condition, an input natural language sentence is tokenized and tallied into a TF vector (IDF is not calculated here, as it already factors into the training document weight). This row vector is normalized by its magnitude. Multiplying this $1 \times |V|$ vector by our $|V| \times N$ TF-IDF matrix yields a $1 \times N$ similarity scores $\in [0, 1]^N$, between our test sentence and each training document.

At first, we averaged these scores to yield an overall score for the condition. However, to capture information on training documents that provide more insight on the composite score than others, we thought of applying a linear map (weighted average) between this $1 \times N$ vector and the final score. Extending this even further, we found greater success in training a Keras model to map these N scores, derived using traditional NLP frequency methods, to a final similarity score.

The training set of the network was the training corpus itself, with associated labels. The model would multiply TF vector for each training document with the TF-IDF matrix of the entire training corpus (including the document itself), and these N similarity scores would be the input vector to the model. The architecture we settled on had a hidden layer of 256 nodes, which outputted a score through a single node with sigmoid activation. This structure outperformed deeper model architectures, indicating the problem space was not very complex. By calculating loss as the cross-entropy between the outputted value and the training label, we trained the model on similarity scores until the F-β scores reached 0.97 with $\beta = 1.75$.

6 Results

6.1 Training Results

	Anxiety	Stress	Depression	Suicide
Accuracy	0.9909	0.9793	0.9888	0.9842
F-β score	0.9514	0.9709	0.9710	0.9709
Val Accuracy	0.9828	0.7136	0.9288	0.8625
Val F-β score	0.9131	0.7808	0.8156	0.8866

The results of our model training are summarized in the provided metrics for anxiety, stress, depression, and suicide detection. Overall, the models achieved high accuracy and F-β scores on the training sets, indicating their ability to learn and classify text related to mental health issues. However, it's essential to note the performance drop on the validation sets, particularly for stress and suicide detection. This suggests a potential challenge in generalizing the models to new and unseen data. The high accuracy and F-β scores on the training sets may be attributed to the models overfitting to the training data. To address this issue, techniques such as regularization and augmentation can be explored.

6.2 Deployment

Notably positive valence ("wow, I'm enjoying life. it's so much fun!")

Detector	Score	Flag
Anxiety	0	-
Stress	1.084×10^{-6}	-
Depression	7.476×10^{-34}	-
Suicide	4.435×10^{-17}	-

Neutral/mild positive valence ("hey, do you wanna play video games later?")

Detector	Score	Flag
Anxiety	0	-
Stress	1.087×10^{-27}	-
Depression	0	-
Suicide	9.635×10^{-31}	-

Mild negative valence ("I have been feeling unwell lately")

Detector	Score	Flag
Anxiety	3.915×10^{-9}	-
Stress	1	"...feeling unwell lately"
Depression	1	"...been feeling..."
Suicide	1.179×10^{-2}	-

Strongly negative valence ("everything is hopeless, nothing works and life is depressing")

Detector	Score	Flag
Anxiety	0	-
Stress	1	"...hopeless, nothing..."
Depression	1	"everything is..."
Suicide	1	"...life is..."

When testing the model on natural language input, we observed expected results. Positive- and neutral-valent statements like "Wow, I'm enjoying life. it's so much fun!" and "Hey, do you wanna play video games later?" fetch no condition-specific similarity scores greater than 1.08×10^{-6}. When tested with negatively charged sentiments, such as "I have been feeling unwell lately" and "everything is hopeless, nothing works and life is depressing", the model flags stress and depression markers, highlighting statements like "feeling unwell lately" and "hopeless, nothing", respectively. The suicidal ideation model requires further tuning, as it seems to flag any life-related statement, e.g. just the bigram "life is". Lastly, the anxiety model seems to have overfit to specific examples of anxiety than words that commonly express the sentiment of anxiety, as it failed to flag any of the negatively-charged statements. Further work should conduct more comprehensive testing, using a large manual dataset or a corpus of texts labeled by a trusted sentiment analysis model.

7 Conclusion

As the use of social media continues to increase, so does the rise in mental health issues. Our goal for this project was to create a model which could try and predict mental health issues based on posts or tweets to see if we could identify signs of these mental health issues earlier.

Our approach relied on our ensemble structure that allowed each model to focus on specific conditions instead of attempting to tackle mental health challenges as a whole. This approach worked, as the deployment examples highlight that different sub-models captured different information from the test sentence.

The provided flagging results underscore the model's potential utility as a red flag detector, assisting mental health providers in identifying individuals who may require intervention. However, ongoing refinement and validation on diverse datasets are essential to ensure the model's robustness and generalizability.

References

Anxiety and Depression Dataset. https://www.kaggle.com/code/docxian/anxiety-and-depression-text-analytics/input.

Huikai Chua, Andrew Caines, and Helen Yannakoudakis. 2022. A unified framework for cross-domain and cross-task learning of mental health con-

ditions. In *Proceedings of the Second Workshop on NLP for Positive Impact (NLP4PI)*, pages 1–14.

Human Stress Dataset. https://www.kaggle.com/datasets/kreeshrajani/human-stress-prediction.

Loke Pak-Yen. Mental health with hugging face pre-trained models. https://www.kaggle.com/code/pakyenn/mental-health-with-hugging-face-pre-trained-models/notebook.

National Institute of Mental Health (NIMH). Mental illness. https://www.nimh.nih.gov/health/statistics/mental-illness.

Student Depression Dataset. https://www.kaggle.com/datasets/nidhiy07/student-depression-text.

Suicidal Tweets Dataset. https://www.kaggle.com/datasets/aunanya875/suicidal-tweet-detection-dataset.