Title
Applications of Natural Language Processing for Predicting Self-Harm Risk

Author
Mori, Yuji

Publication Date
2022

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Applications of Natural Language Processing for Predicting Self-Harm Risk

A thesis submitted in partial satisfaction

of the requirements for the degree

Master of Applied Statistics

by

Yuji Mori

2022

ABSTRACT OF THE THESIS

Applications of Natural Language Processing for Predicting Self-Harm Risk

by

Yuji Mori
Master of Applied Statistics
University of California, Los Angeles, 2022
Professor Frederic Schoenberg, Chair

Self-harm is a subset of mental health that is considered a severe condition requiring immediate attention. This research aims to predict individuals' risk of self-harm using their social media history. This dataset and broader task were originally developed by the eRisk lab at the Conference and Labs of the Evaluation Forum (CLEF). By analyzing the text corpus, it is possible to identify writing patterns that are highly correlated with self-harm. Various methods rooted in Natural Language Processing (NLP) are explored to this end, including sentiment analysis, random forest classification, and deep learning classification using BERT. The results show that adequate classification is attainable with these methods, but the potential to incorporate additional processing steps and model features to increase predictiveness is also discussed.

The thesis of Yuji Mori is approved.

Yingnan Wu

Michael Tsiang

Frederic Schoenberg, Committee Chair

University of California, Los Angeles

2022

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Background

Mental health problems are a widespread issue, yet they are often undiagnosed and untreated. In more severe cases, individuals attempt physical self-harm, which may be preventable with early detection. Information from online channels are regularly used to predict user behavior, and existing research demonstrates the applied use of these techniques in mental health studies. Specifically, an individual's social media posts are a powerful first-person indicator of conditions such as depression and self-harm. The prevalence of these digital communication tools yields a robust collection of written text that greatly enhances traditional research methods.

1.2 Natural Language Processing

The field of Natural Language Processing (NLP) offers many techniques for developing classification models on text data. Traditional methods are based on a bag-of-words representation, where each text is tokenized and the frequency of each token is used as a feature. The goal of this procedure is to convert each text sample into an encoded vector. The resulting vectors can be assembled into matrix representations such as the Document-Term Matrix (DTM), where the columns span the entire vocabulary set of the corpus. This matrix may be further transformed and used in downstream analysis such as predictive modeling.

One weakness of the bag-of-words method is that it cannot capture the context of words, or its role in a sentence as a whole. Word embeddings are a popular method of capturing the relationship between closely related words by representing each word as numerically similar vectors. These embeddings are trained on large corpora, and they may be applied to a multitude of different tasks, with classification being just one of many. Embeddings such as Word2Vec [MCC13] and GLoVe [PSM14] have changed the landscape of NLP research, but transformer-based architectures are especially prominent in more recent state-of-the-art applications.

Bidirectional Encoder Representations from Transformers (BERT) [DCL18] and its variations are widely used in NLP tasks. BERT has become favored over other sequence-based deep learning architectures (such as LSTM and RNNs) due to its bi-directional nature, ability to capture positional encodings, and mechanisms of self-attention and multi-head attention. The model can be used directly to obtain pre-trained word embeddings, and/or it may be fine-tuned for downstream tasks such as classification.

In this paper, the first analysis portion focuses on statistical methods rooted in term frequency and sentiment analysis via pre-labeled unigram lexicons. The next phase uses these document-term matrix inputs to train a predictive machine learning model, where the random forest algorithm is implemented. The final phase uses BERT word embeddings to train a neural network classifier. The predictive performance of the models are evaluated and compared based on several binary classification metrics. In the concluding remarks, the pros and cons of each method are discussed in the context of efficiency, scalability, and other considerations for applied use.

CHAPTER 2

Data Collection and Text Processing

2.1 Data Collection

The data is sourced from an original research publication titled, "A Test Collection for Research on Depression and Language Use" by David Losada and Fabio Crestani [LC16]. The original purpose of this data was to be used in the CLEF research conference under the shared task, "Task 2: Early Detection of Self Harm". The collection contains text data from various internet users, where each post is labeled with a timestamp. The files are organized as a series of XML files, where each file corresponds to a single unique user. Each user has multiple social media posts containing the text comment body and associated metadata.

In total, the data collection contains 1,448 distinct .xml files. Each file represents a single individual, known as a "subject" in the context of this data. Each file is labeled according to their subject number: `subject[N].xml` ,where [N] is to be replaced by an identifying number. The subjects were manually labeled by the original curators of the dataset, where they identified users who explicitly mentioned a medical diagnosis for self-harm risk (as opposed to self-reported or speculated diagnoses).

Within each file, there are one or more records (posts or comments by a single individual). Each record has four XML tags: "TITLE", "DATE", "TEXT", and "INFO". The title field is empty if the record is a comment. The date field is always populated in the YYYY-MM-DD HH:MM:SS datetime format. The text field represents the text body of a post, which may or may not be populated for any given record. Finally, the info field is always populated

with the same string, "Reddit Post" and therefore it is left unused. The dataset spans a history of up to 12 years (from 2008 to 2020), although this range is dependent on each subject/individual. Combined, the entire collection of .xml files yields a total of 746,118 posts and comments.

Finally, the data collection also comes with a separate text file associating each subject with their self-harm label. The file contains one line for each subject and two columns, where the first indicates the subject number, and the second is a binary variable (0/1) representing self-harm status. Among the 1,448 subjects in the data, 152 of them have been labeled with a history of self-harm according to this text file, while 1,296 do not have self-harm history.

2.2 Data Consolidation

In order to extract the information from the 1,448 .xml files, the `xml2` R package is used. The DATE, TITLE, and TEXT fields are read from each file, and this process is iterated over a loop while appending the results to a master data frame object. The names of each subject are also recorded into a new column. Once all the files are read, the subject name matchkey is used to join the self-harm label from a separate flat text file. The result is a data frame with five columns: Title, Date, Text, and Risk Category label. Each row of the data frame represents a single record (either a comment or post). In the context of a corpus, each row represents a single document. However, there are alternative ways of representing this data, one example being the treatment of each subject as its own document. This would increase the document sizes for each user, but in turn would reduce the total number of documents needed to be processed. Further discussion on data representation is presented in the model development chapter (Chapter 4).

2.3 Data Cleaning and Normalization

2.3.1 General Transformations

There are several steps in data cleaning that are common to text mining analysis. These include:

- Convert to Lowercase
- Remove Punctuation
- Remove Numbers
- Remove English Stop Words
- Stemming

The purpose of these transformations is to simplify the overall corpus and identify general patterns across documents in the exploratory analysis phase (Chapter 3). The analysis is focused on calculating frequencies at the individual token level, and the original sentence structure is not preserved. Consequently, many grammatical components of a typical English sentence are no longer relevant. For example, differences due to capitalization and stemmed/unstemmed forms of the same word may be ignored. Similarly, common tokens without any associated sentiment (such as numbers and stop words) are also treated as noise.

However, these transformations are only selectively applied in the predictive modeling phase (Chapter 4). Notably, context-dependent models generally do not require any sort of transformation on the original texts; these tasks are handled by the preprocessing and encoding layers that are built into the network architecture. Therefore, the text inputs for the BERT model in this paper omit the last two transformations (removal of stop words, and stemming). Conversion to lowercase, removal of punctuation, and removal of numbers are still all performed because they are assumed to have minimal impact on the prediction

task. The random forest model, on the other hand, is built with all of these transformations applied.

2.3.2 Removing Duplicate Records

It is often the case on social media for users to post the same content across different channels. On the Reddit social media site, discussion boards are divided into separate channels called subreddits, and users are free to post across these various boards. In more prolific cases, these repeated posts may be considered spam. Therefore, any records that had 10 or more occurrences by the same subject were removed from the data. The 10 post cutoff was manually selected based on the understanding that human users may repeatedly post identical text for promotional reasons, but excessive submission is indicative of automated bot behavior. The core assumption is that these instances do not contribute to the likelihood of self-risk, and their presence pollutes the dataset. However, it may also be possible that this sort of behavior on social media could be a relevant risk predictor. While such analysis is not performed in this research, the flag may be added as an additional model feature.

2.3.3 Removing URLs and other References

In any social media site, a significant proportion of shared posts will include links to other websites, or links to different discussions on the same site. While the contents of these links may be useful, they are removed from this analysis. To accomplish this cleaning step, several regular expression pattern matches were developed to capture the URL patterns using the available functions in the `stirngr` package.

On Reddit, there are many references to other users, which are formatted with the prefix `/u/[username]`. Once again, there are many single-word posts that only mention a username, often because they can be used to invoke bot accounts with a specific purpose. Therefore, it is necessary to find and remove these instances as well. Finally, Reddit also

allows for Markdown-style syntax, so common markdown keywords and symbols were also removed.

Open-source software packages for cleaning Reddit data are available, and other research on this eRisk task has shown success with these libraries [CMA22]. While these resources are not used in this paper, they may be useful for the purpose of reducing time needed to clean these texts.

CHAPTER 3

Text Analysis and Statistical NLP

3.1 Tokenization and Frequency Analysis

Upon cleaning the corpus, the next step is to tokenize the words and analyze their frequencies. Tokenization for this exploratory phase is achieved by using the `unnest_tokens()` function in the `tidytext` R package. The result of this function creates a vector of nearly 7.5 million tokens from the corpus. However, this count includes many tokens with a character length of 1. Therefore, those are filtered out in the downstream analysis. As a result, the self-harm group has approximately 22,300 distinct tokens, while the no self-harm group has nearly 190,000 distinct tokens.

The top words for each group are visualized with wordclouds created by the `wordcloud2` package in R. Figure 3.1 represents the top frequent words for the self-harm individuals. A key characteristic of this figure is the presence of words with strong sentiment association in English, which includes words such as "feel", "bad", "life", "hate", and others. Meanwhile, the wordcloud for the non self-harm group (Figure 3.2) does not contain these words, and it can be said that the presented tokens are neutral in terms of sentiment.

Figure 3.1: WordCloud for self-harm = 1 (yes risk)

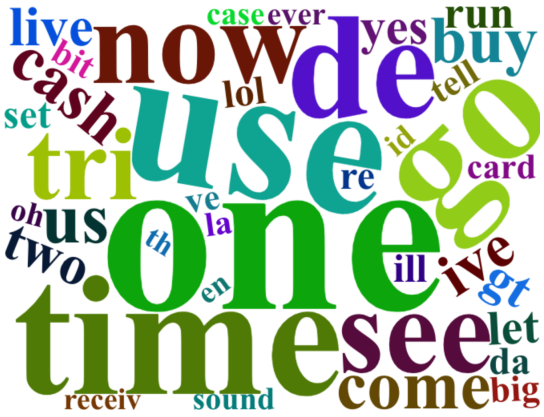

Figure 3.2: WordCloud for self-harm = 0 (no risk)

3.2 Token Importance and TF-IDF Calculations

Term-Frequency Inverse Document Frequency (TF-IDF) is a measure used to assess relative importance of a token, word, phrase, etc. within the context of a collection of documents in a corpus. For each risk group, the top 10 tokens by TF-IDF are calculated (Table 3.1 and Table 3.2) and assessed in the context of self-harm. The list of tokens for the self-harm group do not appear to be normal English words, so the results of this TF-IDF calculation are difficult to interpret. These tokens appear to be specific terminology that does not apply

to self-harm. On the other hand, the no self-harm group once again exhibits neutral words that do not pertain to self-harm. The table includes many tokens related to finance, such as “crypto” or “cryptocurr”, both referring to the topic of cryptocurrency. It is likely that certain discussions about finance are rare in the corpus, which is why they appear in the list.

word	n	tf-idf
styro	51	2.18
harmer	25	2.18
avpd	24	2.18
wurt	21	2.18
clonazolam	19	2.18
merm	18	2.18
tegu	16	2.18
tucut	14	2.18
eysoreofsocieti	13	2.18
diclazepam	12	2.18

Table 3.1: Top 10 Tokens by TF-IDF (self-harm)

word	n	tf-idf
crypto	10348	0.0244
chim	8008	0.0244
blockfi	5626	0.0244
cryptocurr	4014	0.0244
sofi	3478	0.0244
crypterium	3279	0.0244
signup	2864	0.0244
ethereum	2776	0.0244
usdc	2494	0.0244
webul	2342	0.0244

Table 3.2: Top 10 Tokens by TF-IDF (no self-harm)

3.3 Correlation Analysis

After identifying keywords, a correlation analysis is performed between the tokens associated with each group. Specifically, the proportion between a given word and their presence in each group is calculated, then those proportions are compared.

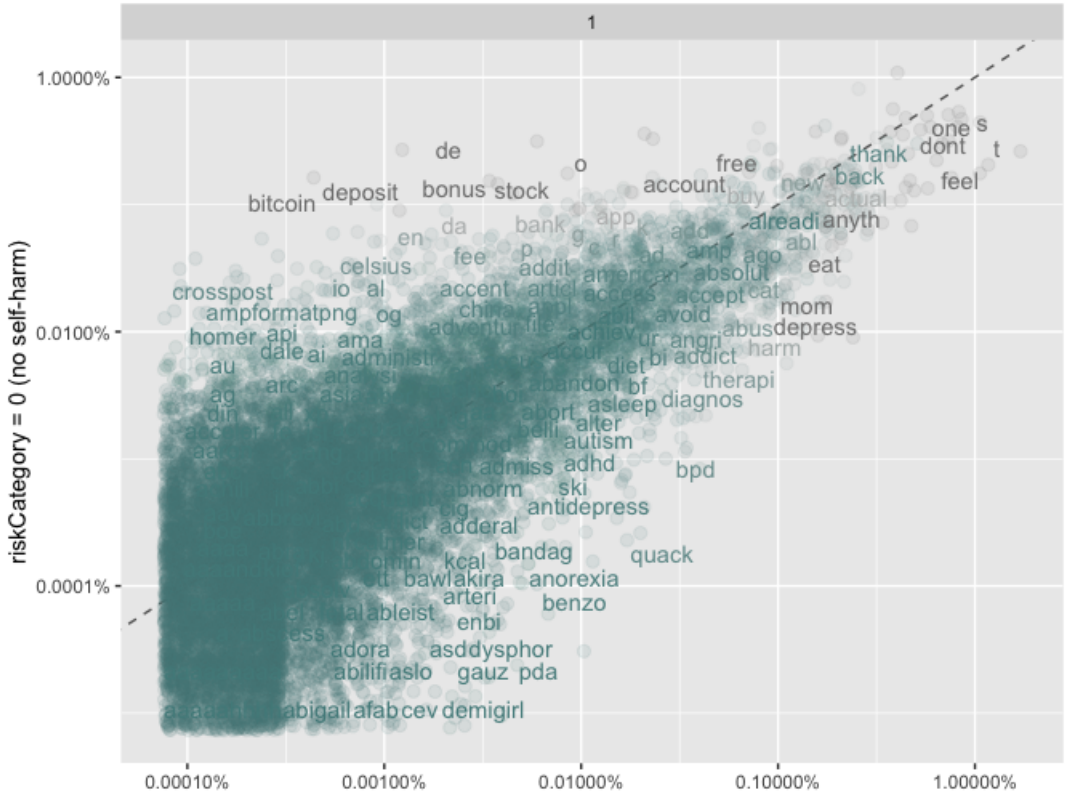

Figure 3.3: Comparison of Token Proportion by Risk Category

Running a Pearson's Correlation test, the two groups have a correlation of 0.86, which implies a high degree of similarity for the relative proportions of common words in each group. The results can also be plotted onto a graph (Figure 3.3) where the two proportions for each word fall on the x-axis (for proportion in self-harm) and y-axis (for proportion in no self-harm) respectively. The words that are more unique to the self-harm group appear in the bottom-right quadrant of the plot. Once again, there are words associated with sentiments like "depress", "diagnosis", "therapi", and others. The upper right quadrant represents words that are more unique to the no self-harm group, which once again has many references to neutral, finance-related terms.

3.4 Sentiment Analysis on Tokens

Basic sentiment analysis is also performed to label the tokens in each group with an associated sentiment rating. Note that this process does not refer to sentiment analysis over an entire sentence or post, but rather it is simply labeling individual tokens based on a pre-existing sentiment dictionary. There are many ways to assign sentiment to a token, but due to the binary nature of this dataset, the most intuitive procedure is to divide the token into "positive" and "negative" sentiment labels. For this purpose, the `bing` library is used, which is a dictionary of pre-labeled words from the `tidytext` package in R. Many tokens in the corpus are not present in this pre-labeled dictionary, so many terms from each group are dropped. The terms that remain have an associated positive/negative label, and these frequencies can be plotted and compared between groups (Figure 3.4).

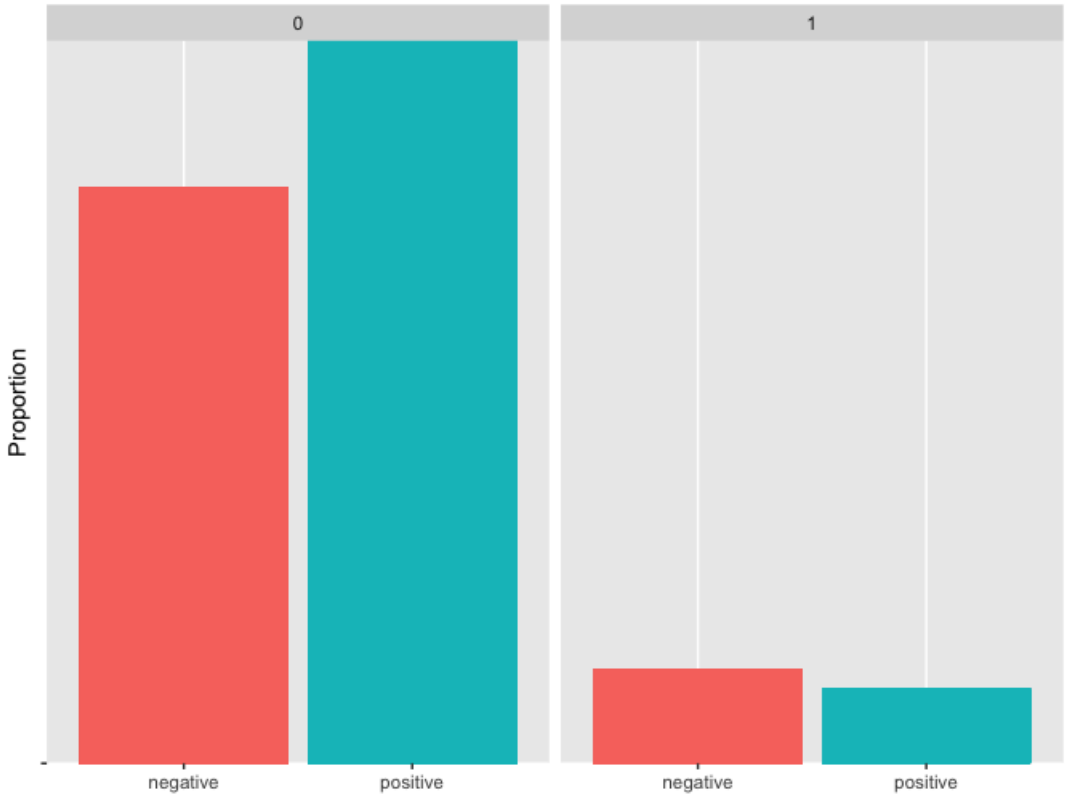

Figure 3.4: Sentiment Analysis using bing Dictionary (No Self-Harm vs Self-Harm)

The bar plots show that the No Self-Harm group has a much higher proportion of positive terms than negative, although they are both significantly represented in the collection of documents. The Self-Harm group, while having fewer terms, have slightly more negative instances than positive ones. This pattern is in line with expectations, considering that self-harm may be assumed to indicate an emotionally negative state. These results support the idea that even a simple classifier, based on the presence/absence of terms, may be sufficient to predict the risk of self-harm. Prior studies show that these sentiment labels can be directly used as a classification flag in conjunction with token frequency predictors [KRN20].

CHAPTER 4

Classification Models for Predicting Self-Harm Risk

The goal of this research is to create predictions on self-harm risk using this data in order to provide support to those afflicted with this condition. While there are many statistical methods available for building such a classifier, machine learning implementations are the most prevalent. As a preliminary model, the Random Forest algorithm is selected due to its reliability as an ensemble method, as well as having high explainability for its decision tree structure. Previous studies have successfully used similar algorithms for classifying mental health diagnoses using text data [AA17] [BG21], including random forest, support vector machines (SVM), and gradient boosting. However, most modern NLP models rely heavily on deep learning architecture, which convert text into word embeddings and feed these vectors through sophisticated neural networks. Based on existing research on the eRisk data [PML21], a deep learning model is implemented using fine-tuned BERT. Then, these models are compared to assess for any performance lift.

4.1 Design Adaptations from CLEF eRisk Shared Task

In this research, several changes are made to the original eRisk shared task prompt and evaluation process. Some of these adjustments are necessary due to inherent changes in the dataset since the CLEF 2021 conference, while others are decisions relating to experimental design. These differences should be considered when comparing any results with existing literature based on this dataset.

The first change pertains to the overall dataset. In the 2021 eRisk Conference, participants had access to specific training and test data subsets created by the conference organizers (Table 4.1). Meanwhile, the data acquired for this paper consists of only the 1,448 test set subjects from the original conference. As a result, the data partitions differ from the original. The splitting criteria is further discussed in the next section.

	Train		Test	
	Self-Harm	*Control*	*Self-Harm*	*Control*
Num. subjects	145	618	152	1296
Num. submissions (posts & comments)	18,618	254,642	51,104	688,823

Table 4.1: Summary Statistics of eRisk 2021 Data [PML21]

The next adaptation is a simplification of the data structure. The goal of the predictive models is to predict the risk of self-harm for each *subject*. Therefore, it is reasonable to concatenate each subject's individual posts into a single string, as opposed to treating each post as a separate classification problem. For example, if an individual with positive self-harm risk has 10 social media posts, those posts are condensed into a single text with a positive self-harm flag. The submissions are concatenated with a single whitespace delimiter, and they are arranged in order of post date, with the earliest items in the beginning of the combined string.

One consequence of this strategy is that the date field associated with each post is no longer used. While a sequential order is still retained by concatenation procedure, a single date cannot be assigned to the combined text for a given subject. It follows that several time-dependent evaluation metrics developed by the eRisk authors can no longer be calculated. These metrics include: Latency, Speed, Latency-Adjusted F1 Score, and Early Risk Detection Error (ERDE). Therefore, any predictive model evaluations will focus on standard metrics such as Precision, Recall, and F1 Score.

4.2 Splitting the Data

Both the random forest model and the BERT model are trained and evaluated on the same data partition. The data is split by holding out one-third (33%) of the observations as a test set, while the remaining two-thirds (67%) are used for model training. Due to the imbalance in the number of samples between the risk and no risk group, the split is also stratified on the risk category field. The split is performed after any preprocessing steps or transformations that are specific to each model. These precursory steps eliminate one test subject from the overall sample in Table 4.1, reducing the total observations from 1,448 to 1,447.

The new partitions are shown in Table 4.2. In total, 1,085 subjects are used in the training process, while the remaining 362 subjects are reserved for the test phase.

	Train		Test	
	Self-Harm	*Control*	*Self-Harm*	*Control*
Num. subjects	113	972	38	324

Table 4.2: Train and Test Partitions for Random Forest and BERT Models

4.3 Random Forest Classification Model

4.3.1 Removing Sparse Terms

The Random Forest model treats each unique word in the corpus vocabulary as an input feature, where the input value corresponds to the term frequency within a given social media post. The classifier uses all terms as predictor variables, which quickly leads to issues with computational load. In addition, the presence of too many uncommon terms in a decision tree classifier may easily lead to overfitting. Therefore, a general procedure is to remove sparse terms from the corpus before training a classification model. The complete document-term matrix consists of 1,447 documents and 279,320 terms. Only keeping the top 99 percent of

words by document presence, the number of input terms is reduced to 14,139 variables.

4.3.2 Model Training

The `randomforest` package in R is used to develop the random forest model. The parameters are manually tuned in several ways in order to improve predictiveness, minimize computation time, and prevent overfitting. The maximum number of trees is set to 500 and the maximum number of terminal nodes to 500 in order to control the depth of each random tree. The minimum node size is set to 5, which may be large for this small dataset, but the goal is to prevent overfitting. The observations are weighted (or balanced) in order to account for the over-representation of the negative class in the dataset. The splitting criteria in this implementation uses the Gini impurity metric.

4.4 BERT

4.4.1 Data Preparation

Several adjustments were made in the data preparation methodology for input into the BERT classifier model. In general, pre-trained word embedding representations are meant to handle text data with minimal preprocessing or modification of the original sentence structure. In the preceding analysis, modifications such as the removal of capitalization, punctuation, numbers, etc. were made to the data. However, these treatments may be detrimental to the BERT model's performance, as they may be crucial components to the true meaning of the sentence. Specifically, the process of removing stop words, stemming words, and removing sparse terms had the largest impact on the original sentence structure. Therefore, these three steps will be omitted from the data preparation step.

4.4.2 Model Architecture and Training

The specific BERT model used in this research is the uncased BERT base model. The model uses *L=12* hidden layers (encoder blocks), a hidden size of *H=768*, and *A=12* attention heads. Other variations of BERT are available, and they vary in the overall size of model as well as the training techniques used to obtain their weights. Existing research shows that newer iterations of BERT achieve slightly better performance on self-harm risk data and similar datasets [GGY21]. Despite these results, the base model is selected for this paper in order to establish more direct comparisons against the baseline random forest classifier, as well as to minimize runtime in the model development phase.

The texts are first passed through a preprocessing model, which is also developed by the authors of BERT and available on the TensorFlow Hub repository. The encoder layer converts all text into specialized BERT embeddings. BERT only accepts text inputs of up to 512 tokens in length, so any observations with a higher length are automatically truncated during these two steps. A dropout layer is added for model stability, and a final dense layer with a sigmoid activation function is used to output class probabilities for the binary classification problem. The full model is built using the TensorFlow library in Python, and the basic network diagram is shown in Figure 4.1.

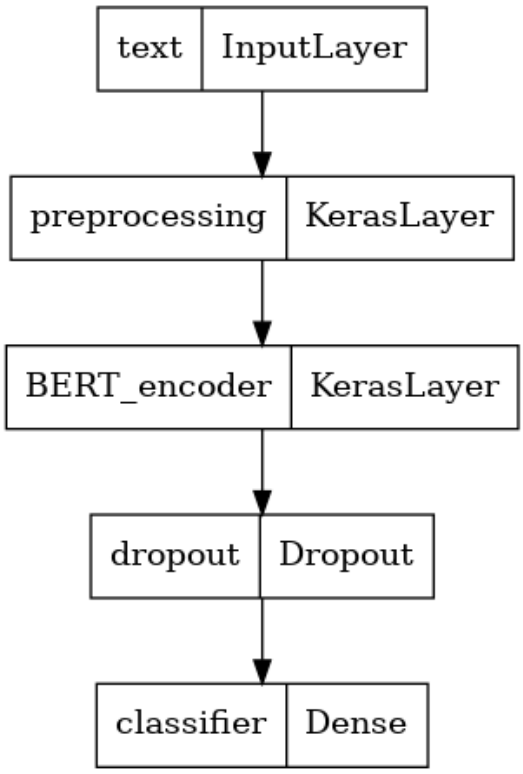

Figure 4.1: TensorFlow Model Structure using BERT Encoder for Classification

The training process uses the Adam optimizer, and the loss function is based on binary cross entropy. The model is trained over 10 epochs and a batch size of 32. Similar to the random forest model, class weights are assigned to each observation to balance the input data.

4.5 Model Results

The performance of both models are evaluated on the basis of Precision, Recall, and F1 Score. The test set consists of 362 subjects in total, which are classified into either the positive (self-harm) or negative (non self-harm) class. The evaluation metrics are compiled and shown in Table 4.3.

	θ	F1	Precision	Recall
Random Forest	0.161	0.538	0.424	0.737
BERT	0.565	0.588	0.532	0.658

Table 4.3: Evaluation Results for Random Forest and BERT Models

The θ column represents the best probability threshold used to classify the test subjects. This optimal point is determined by generating ROC curves and identifying where positive-class classification is maximized. In the case of the BERT model with sigmoid activation, the output layer directly computes the class probabilities for each observation. In the random forest algorithm, each decision tree in the ensemble produces a predicted class label (as opposed to a probability), so the probability of self-harm is derived directly from the proportion of positive class outcomes in the ensemble.

With these thresholds, the BERT classifier, as expected, outperforms the random forest classifier in all metrics except for Recall. Due to the small sample size of the test set (containing only 38 positive self-harm subjects), the actual classification results only differ by a few subjects. Therefore, despite its overall lower performance, the random forest model exhibits results that may be considered competitive. It may even be argued that Recall is the most important metric to maximize, given the consequences of a false negative result (i.e. a self-harm subject who remains undetected). The F1 score of 0.588 for BERT is comparable to results previously obtained in previous eRisk conference iterations [PML21].

The classification results for each model are presented in a confusion matrix representation below (Table 4.4 and Table 4.5). The columns represent the truth labels, while the rows represent the predicted classes.

		Prediction	
		False	True
Actual	False	286	38
	True	10	28

Table 4.4: 2x2 Confusion Matrix for Random Forest Classification Results

		Prediction	
		False	True
Actual	False	302	22
	True	13	25

Table 4.5: 2x2 Confusion Matrix for BERT Classification Results

Lastly, feature importance values are also calculated. The random forest algorithm is capable of measuring the importance of each individual token as it relates to the splitting criteria. Because the model is trained on the Gini impurity algorithm, the feature importances are represented by Mean Decrease in Gini Impurity. The top 25 tokens based on this metric are shown in Figure 4.2. It is evident that many of these terms are directly related to self-harm and other mental health conditions. Several of these tokens are also found on the correlation plot created in the exploratory analysis phase (Figure 3.3). Due to their high correlation with self-harm risk, the presence/absence of these individual tokens may be inferred to be the most important features of the BERT classifier as well. If the benefits of BERT embeddings only have a minor contribution to the model performance, then these results also explain how the random forest model yields performance metrics that are comparable to the BERT model.

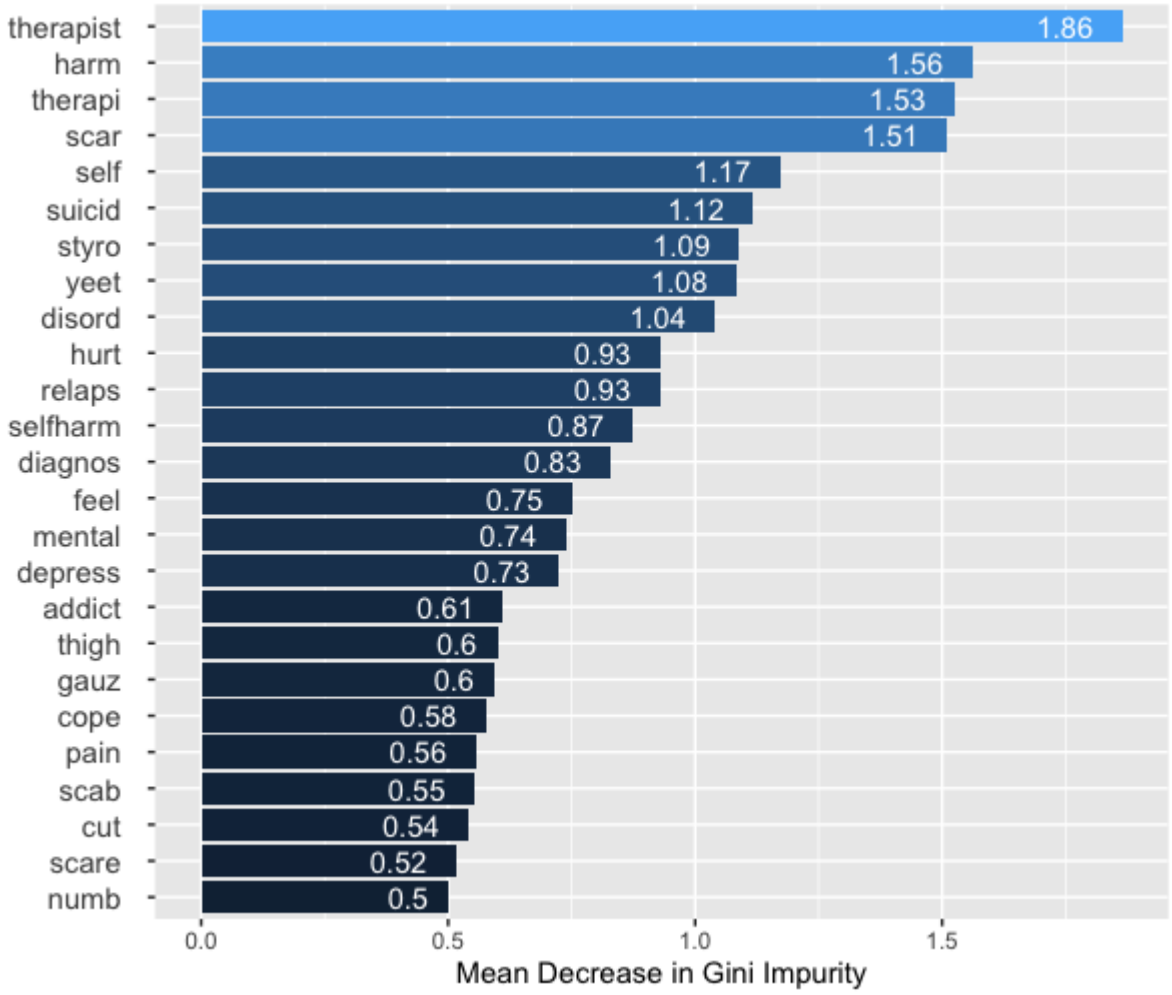

Figure 4.2: Top 25 Tokens by Feature Importance in Random Forest

CHAPTER 5

Conclusion

The findings illustrate that the BERT model yields a significant performance lift on the self-harm risk classification task relative to the random forest model. In the context of serious conditions such as self-harm, the importance of achieving a high prediction rate can have great implications in terms of clinical outcomes.

In practical settings, there are considerations surrounding the complexity of the models. Firstly, a more complex model results in increased computation times, which may present issues when delivering these insights. While the architecture of the BERT model is undoubtedly more complex than the random forest model, the computation time for this particular model is minimized due to its reliance on pre-trained weights and simple design of the final dense layer. Next, there is the concern of model explainability. For deep learning models in particular, the method of calculating weights is difficult to communicate in an industry context. With this respect, the random forest holds the advantage of producing a clear, traceable decision structure.

Both models also require a significant degree of text cleaning and preparation due to the unrefined nature of social media writings. As shown in the text processing sections (Chapter 2), there are many domain-specific and site-specific patterns that must be handled before passing to any model. While BERT can handle raw texts to some degree, the presence of unformatted URLs, spam posts, punctuation from markdown syntax, and other noisy artifacts may lead to reduced performance, or at the very least an unnecessary increase in model runtimes. Data processing techniques such as the ones implemented in this paper

should therefore be at the forefront of any NLP task on social media data.

Various additional methods may be considered for future model improvement. The main limitations of this paper are centered around the small size of the dataset. While the best approach is simply to collect more data from Reddit with associated self-harm labels, it may also be possible to improve predictions by fine-tuning BERT on adjacent social media (such as Twitter data) which are more widely available. Another point to consider is that non-text features may also be added into these predictive models, such as the number of posts by each subject, the timespan of account activity, and posting velocity. While these features are not in the scope of NLP research, the information may have a relationship with self-harm. Further investigation in these directions may improve self-harm research as well as similar issues that may be detected in social media texts.

REFERENCES

[AA17] Maryam Mohammed Aldarwish and Hafiz Farooq Ahmad. "Predicting Depression Levels Using Social Media Posts." *2017 IEEE 13th International Symposium on Autonomous Decentralized System (ISADS)*, pp. 277–280, 2017.

[BG21] Tanmay Basu and Georgios V Gkoutos. "Exploring the Performance of Baseline Text Mining Frameworks for Early Prediction of Self Harm Over Social Media." In *CLEF (Working Notes)*, pp. 928–937, 2021.

[CMA22] Elena Campillo-Ageitos, Juan Martinez-Romo, and Lourdes Araujo. "UNED-MED at eRisk 2022: depression detection with TF-IDF, linguistic features and Embeddings." *Working Notes of CLEF*, pp. 5–8, 2022.

[DCL18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *CoRR*, 2018.

[GGY21] Yuting Guo, Yao Ge, Yuan-Chi Yang, Mohammed Ali Al-Garadi, and Abeed Sarker. "Comparison of pretraining models and strategies for health-related social media text classification." *medRxiv*, 2021.

[KRN20] E. Rajesh Kumar, K.V.S.N. Rama Rao, Soumya Ranjan Nayak, and Ramesh Chandra. "Suicidal ideation prediction in twitter data using machine learning techniques." *Journal of Interdisciplinary Mathematics*, **23**(1):117–125, 2020.

[LC16] David E Losada and Fabio Crestani. "A test collection for research on depression and language use." In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pp. 28–39. Springer, 2016.

[MCC13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient Estimation of Word Representations in Vector Space.", 2013.

[PML21] Javier Parapar, Patricia Martín-Rodilla, David E. Losada, and Fabio Crestani. "Overview of ERisk 2021: Early Risk Prediction on the Internet." In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 12th International Conference of the CLEF Association, CLEF 2021, Virtual Event, September 21–24, 2021, Proceedings*, p. 324–344, Berlin, Heidelberg, 2021. Springer-Verlag.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "GloVe: Global Vectors for Word Representation." In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.