DETECTING SUICIDE IDEATION/ATTEMPTS WITHIN SOCIAL MEDIA

Timothy William Reitz

A Thesis Submitted to the
University of North Carolina Wilmington in Partial Fulfillment
of the Requirements for the Degree of
Master of Science

Department of Computer Science

University of North Carolina Wilmington

2022

Approved by

Advisory Committee

Gulustan Dogan	Geoff Stoker
Yiyi Yang	Yang Song Chair

Accepted By

Dean, Graduate School

TABLE OF CONTENTS

ABSTRACT

Background: Suicide is a serious problem that affects individuals all over the world. The suicide rate increased by 30% between 2000 and 2018 [1]. Over the past couple of years, suicide has been in the top 10 leading causes of death [1]. In the year 2020, 3.2 million people planned suicide; out of the 3.2 million people, 1.2 million attempted suicides, and 45,979 died by suicide [1]. Suicide survivors and members surrounding survivors are at high risk of developing suicidal ideation [2]. These people often feel complex emotions involving guilt, shame, anger, and denial [2]. This state often breeds ideal grounds for perceiving stigma [2]. According to Cambridge, stigmatization is the act of treating someone or something unfairly by publicly disapproving of him, her, or it [3]. This perceived stigma around suicide is often a barrier for individuals to reach out and seek treatment. According to the American Psychological Association, patients who received early intervention had 30 % fewer suicide attempts [4]. With that being said, numerous researchers have attempted to detect suicide ideation/attempts within social media posts across multiple platforms [2,7-12]. These researchers used a plethora of artificial intelligence algorithms to accomplish their tasks.

Objective: The primary purpose of this study was to develop, compare, and contrast artificial intelligence algorithms to determine which could detect suicide ideation/attempts within social media data. The objective is to gather and annotate data using manual, machine learning, and deep learning algorithms for completeness. Then they were compared to determine which model would give the highest accuracy and best overall confusion matrix while keeping in mind that false positives are more favorable than false negatives.

Method: The data for this study was taken from the subreddit r/SuicideWatch using an application programming interface between 2019 and 2020. For annotating the collected data, various methods were used, including manual coding, Support Vector Classifier, K-nearest neighbor, logistic regression, decision tree, stochastic gradient descent, random

forest, Naive Bayes, and a temporal convolution neural network. As the datacollection continued; these algorithms were trained on posts ranging from a few hundred to 1,500.

Results: At the end of the study, it was determined that the machine learning algorithms performed as well as a human annotator with an accuracy of around 90 %. It was also determined that the scikit-learn Naive Bayes did not outperform the machine learning algorithms, having an accuracy of around 78%. Finally, the non-optimized temporal convolution neural network performed as well as the machine learning algorithms.

Conclusion: The results showed that the temporal convolution neural network was the top performer for detecting stigma with a social media post. This temporal convolution neural network can be used instead of manual coding and perform as well as a human coder, reducing the time needed to annotate. It may prove that future work on the temporal convolution neural network can achieve results higher than a human coder.

ACKNOWLEDGEMENT

First, I would like to thank my chair, Dr. Yang Song. Without him and his guidance, this work would never have been possible. I also would like to thank my committee members, Dr. Gulustan Dogan, Dr. Yiyi Yang, and Dr. Geoff Stoker, for their time, knowledge, and guidance throughout my academic career. Without my committee members mentoring and impact, I would not have been able to expand my knowledge and study of the topic. I would like to extend the deepest thanks to Taylor Griffin and Katherine Scoggins for their contribution to the manual annotation of the dataset. Finally, I would like to thank the GEARS students who provided support during the early stage of this study. These GEARS students played a vital role in the development of the active machine learning portion of this research.

DEDICATION

I would like to dedicate this thesis to my mother, who passed away in 2016. Throughout my time with her, she instilled principles and values that have shaped who I have become. She is no longer with us, but her impact on the world and myself is still seen. Without her hard work and determination to see me succeed, shown in her parenting, I would not be where I am today. I dedicate this thesis to her for all the reasons above and many more.

LIST OF TABLES

LIST OF FIGURES

1 Introduction

1.1 Motivation

Suicide is a serious problem affecting more than 1.4 million people directly and indirectly [1]. Between 1999 and 2019, suicide rates increased [1]. The suicide rate increased by 30% between 2000 and 2018 [1]. Over the past couple of years, suicide has been in the top 10 leading causes of death [1]. In 2020, 3.2 million people planned suicide; out of the 3.2 million people, 1.2 million attempted suicide, and 45,979 died by suicide [1]. Suicide survivors and members surrounding survivors are at high risk of developing suicidal ideation [2]. These people often feel complex emotions involving guilt, shame, anger, and denial [2]. This state often breeds ideal grounds for perceiving stigma [2]. Suicide affects age groups differently, with it being skewed toward the younger populous. Suicide is the second leading cause of death among 10 - 35-year-olds. [1]. Interestingly this is the same age group that uses social media the most [5]. This study will use a data set from the subreddit r/SuicideWatch, which will capture a great number of people being affected by suicide.

According to Cambridge, stigmatization is the act of treating someone or something unfairly by publicly disapproving of him, her, or it [3]. Many people have stigmas about suicide that include but are not limited to seeing suicide as a selfish act, a way to escape, a sign of cowardice, and a revenge act [3]. Putting this stigma on suicide reduces the rate at which individuals seek available early intervention [4]. These stigmas also change how society and individuals perceive themselves. This stigma is often a barrier for individuals to reach out and seek treatment. According to the American Psychological Association, patients who received early intervention, such as self-report safety plan, and Coping Long Term with Active Suicide Program, had 30% fewer suicide attempts [4].

Previously, many researchers have attempted to detect stigma in social media across many platforms. These researchers used a plethora of artificial intelligence algorithms to

accomplish their tasks. They also used various sizes of data sets ranging from 974 to 2.35 billion. Challenges have been faced when detecting stigma due to the complexity of languages, new and changing stigmas, and obtaining and annotating the data.

1.2 Objectives

The first objective of this study was to develop an algorithm to interact with the Reddit application programming interface to collect data from the subreddit r/SuicideWatch. The second objective of this study was to develop a baseline using human coders to annotate the data. To ensure inter-coder reliability, the Cohen-Kappa score was used. The third objective of this study was to create a voting classifier to achieve results similar to a human coder. Once desired results were achieved, this study's final and primary objective was to develop a traditional machine learning and temporal convolution neural network model to achieve scores higher than the voting classifier used in the previous objective.

1.3 Research Question

Based upon previous work with suicide-related detection within social media posts, we know that using artificial intelligence and machine learning has provided high accuracy in annotating data. Most previous studies needed an Oracle to provide insight to ensure the accuracy of the algorithm can maintain a level equal to a human coder. Merriam-Webster defines an Oracle as a person giving wise or authoritative decisions or opinions [6]. A question intended to be answered by this study: Can a deep learning algorithm outperform previous machine learning with suicide-related detection? Another question designed to be answered by this study was whether a custom deep learning algorithm could outperform pre-built deep learning algorithms with suicide-related detection? To answer these questions, a baseline was established by a human coder. Then a machine learning algorithm was developed to achieve results similar to a human coder. After both a scikit-learn Naive Bayes model and temporal convolution neural network were coded. They were compared not only

to each other but to the previous machine learning algorithms as well.

1.4 Research Scope

For this study, multiple algorithms were applied to the dataset, which is elaborated on in later sections of the paper. The dataset is limited to the subreddit r/SuicideWatch and the years 2019 and 2020. Even though these constraints bound the study, the algorithms and methods can be applied to other datasets and algorithms.

1.5 Paper Organization

This thesis has been organized into sections: Introduction, Related Works, Dataset, Annotation, Active Learning, Deep Learning, Results, Conclusion, and Future Work. Discussed in the introduction is the motivation of the research, the objectives to be covered, the research questions to be answered, and the scope of the research. In the related works section, similar previous works are discussed, along with how this study has similarities and differences to those earlier works. The following section, titled Dataset, discusses where and how the data was collected for the study. The Annotation section describes how the collected data was annotated to create a labeled dataset to be used in the training of algorithms discussed in sections 5 and 6. Also addressed in the Annotation section is how inter-coder reliability was maintained. Following the Annotation section, the Active Learning section discusses three main topics Data Cleaning, Machine Learning Algorithms, and Machine Learning Method. The Data Cleaning section explains how the data was prepped for machine learning. The section labeled Machine Learning Algorithms defines the background of the machine learning algorithms. The methods which were used to annotate the data using active machine learning were discussed in the sections labeled Machine Learning Method. The following section, labeled Deep Learning, has three subsections: Deep Learning Algorithms, Naive Bayes method, and Temporal Neural Network Method. Within the Deep Learning Algorithms section, the

methods that were used are discussed. How the Naive Bayes was used to annotate the dataset based upon the data cleaned in the Data Cleaning section was described in the Naive Bayes Method section. The Temporal Neural Network section discussed the method used to create, train, and annotate the cleaned data. Next, the Conclusion section discusses the research results and details how they were obtained. Finally, in the Future Work section, the future applications of this work were discussed.

2 Related Works

The table below is a summary of previous research done. As shown below, several important aspects were focused on, such as the data source, the size of the dataset, the way the dataset was annotated, the algorithm used, and the hypothesis.

Name	Data Source	Size	Annotation	Model	Hypothesis
Planning and Social Media: A Case Study of Public Transit and Stigma on Twitter	Twitter	5,000 tweets from 500 agencies 64,000 comments	Two control groups; celebrities(positive) and villain(negative). Compared transit against public and private agencies(airlines are control group). Final control is social welfare. Both positive and negative comments. Used ML to annotate data. Hand coded a sample to check for accuracy. Two students and author coded 200 comments.	machine-learning algorithm	Do social media users describe transit planning, management, and services in a positive or negative manner? Do differences in social media interactions influence the tone of the discussion surrounding agency services, planning, and public management on social media?
Assessing Suicide Risk and Emotional Distress in Chinese Social Media: A Text Mining and Machine Learning Study	Web-based survey	974 Weibo users participated in the survey	Weibo posts were parsed and fitted into Simplified Chinese-Linguistic Inquiry and Word Count (SC-LIWC) categories.	support vector machine (SVM) model automatically on 5 risk factors.	The aim of this study was to explore whether computerized language analysis methods can be utilized to assess one's suicide risk and emotional distress in Chinese social media.
Machine Learning, Sentiment Analysis, and Tweets: An Examination of Alzheimer's Disease Stigma on Twitter	Twitter	31,150 AD-related tweets collected via Twitter's search API based on 9 AD-related keywords.	Two researchers manually coded 311 random tweets on 6 dimensions. 1% of data to train. 99% of the data to test.	machine-learning algorithm, NLP Toolkit, the WEKA data mining toolkit ,SciKitLearn, and other.	Describe our development of a semi-automated text coding method and use a content analysis of Alzheimer's disease (AD) and dementia portrayal on Twitter to demonstrate its use. The approach improves feasibility of examining large publicly available data sets.
Development of a Behavioral Health Stigma Measure and Application of Machine Learning for Classification	survey instrument was deployed electronically available to participants through SONA Systems	1,904 participants	five-point Likert scale and a binary proxy question.	Data collected during the spring semester used to create an easy-to-administer stigma scale. Machine learning techniques were then applied to the data for the purpose of developing a classification decision tree	the goal of this study was to demonstrate the potential for using machine learning as a tool to analyze patterns of social stigma as a complement to traditional research methods.
Identification of Imminent Suicide Risk Among Young Adults using Text Messages	multi model data set includes personal communication, social media data, web browsing history, and mental health history	2,377 students, limited to only 26 of the participants	Participates label their data collected based on Suicidality and depression. Then again split by Certain and Uncertain.	supervised machine learning to build classifiers that predict a binary classification of depression Two sets of features were used to construct the classifiers	Can text mining identify periods of increasing risk states for suicidality (e.g., depression to suicidality) based on everyday communications?
Stigma Annotation Scheme and Stigmatized Language Detection in Health-Care Discussions on Social Media	novel health-care data set two biggest health-care walls on Facebook. One pro-vaccination(283,274 users), anti-vaccination(224,851 users)	A total of 4,502 comments (2,251 anti- and 2,251 pro-vaccination comments, 8,584 sentences, and 105,470 tokens) were collected from January to March 2018.	each comment was labelled by trained annotators and Amazon MTurk experts three times according to class definitions: stigma, not stigma, and undefined.	TF-IDF, N-grams +Logistic Regression Support Vector Machine Naive Bayes MLP (Multi layer Perceptron) Random Forest K-Nearest Neighbours SGDC (Stochastic Gradient Descent) LSTM BiLSTM CNN fastText 25 Epochs fastText 25 Epochs, N-grams	How to build a rigorous annotation scheme and achieve higher inter-rater agreement when there is no consensus on a concept definition among the researchers? What are the characteristic features of stigmatised language in vaccination comments on social media? Can deep learning models be better predictors of health stigma given the relatively small labelled data set?
Data Mining of Web-Based Documents on Social Networking Sites That Included Suicide-Related Words Among Korean Adolescents	163 social media Web sites in South Korea	2.35 billion posts. final analysis s composed of 99,693 documents	coded into structured data through text mining and opinion mining as follows: 1 for expressions approving of suicide or neutral to suicide and 0 for expressions disapproving of suicide.	crawler model fit using incremental fit indices, normed fit index, comparative fit index, Tucker-Lewis, and absolute fit indices	To investigate online search activity of suicide-related words in South Korean adolescents through data mining of social media-Web sites as the suicide rate in South Korea is one of the highest in the world.
Text mining analysis of teachers' reports on student suicide in South Korea	student suicide case report from the Korean Ministry of Education	417 student suicide case from 2011 to 2017, from the Ministry of Education. 246 male and 171 female students, 288 high school, 103 middle school, and 26 elementary school students.	first carried out a prepossessing procedure on the text data that were encoded in the Unicode Transformation Format 8 bit to eliminate noisy information such as numbers, punctuation marks, and stop words.	R version x64 3.4.2 was used [13], and Korean Natural Language Processing (KoNLP) [14] and National Information Society Agency Dictionary (NIADic) Latent Dirichlet allocation (LDA)	to identify the characteristics of Korean student suicides based on teachers' reports and gain a better understanding of Korean student suicides from the teacher's perspective.

Table 1: Related Works from Previous Studies

Upon researching previously completed studies, many were found that provided a foundation and insight to previously explored options.

As shown in the tables above, we can see that "A Case Study of Public Transit and Stigma on Twitter," "Assessing Suicide Risk and Emotional Distress in Chinese Social Media," "An Examination of Alzheimer's Disease Stigma on Twitter," and "Data Mining

of Web-Based Documents on Social Networking Sites," all use datasets collected from a public social media outlet similar to the origins of the dataset in this study [7–10]. In addition, datasets of previous studies vary in size ranging from 974 to 2.35 billion posts. This study currently has approximately 1.4 million collected posts and comments from the subreddit r/SuicideWatch. However, only about 1600 posts have been annotated so far. This shows that this study is neither the largest at 2.35 billion nor the smallest at 417 but falls somewhere in between. Based on previous studies, the sample size of 1600 did provide a good baseline for this study.

As shown in Table 1, all previous studies used manual and machine coders to annotate the data [2, 7, 9]. Other studies used a combination of Lexicon-based coding, a five-point Likert scale, and having participants self-code [8, 11, 12]. In this study, 350 posts were manually coded by 2 coders, and inter-coder reliability was checked using the Cohen Kappa score, which is discussed in a later section. To increase the dataset to its current size, active machine learning with an Oracle was used. This will be discussed in a future section.

Finally, previous studies have used various machine learning models to predict the presence of the expected stigma. This study explored multiple options in order to determine the best model, not only for active machine learning but for deep learning as well. The last thing to discuss in previous works and this study is that 2 dimensions were predicted to create a pipeline for the other 24 dimensions that followed. The "An Examination of Alzheimer's Disease Stigma on Twitter" research predicted on 6 dimensions, which is the highest so far [9].

3 Dataset

3.1 Data Source

This study's primary source of data was derived from the subreddit r/SuicideWatch. Reddit is a social media platform that contains many different forums where registered users can post their thoughts and comment on others. These posts can then either be voted up or down. To date, Reddit has 430 million active users who engage in their content at least once a month [13]. Reddit was first created in 2005 by Steve Huffan and Alexis Ohanian [14]. The subreddit r/SuicideWatch was created 3 years later, in 2008 as a place where people could express their emotions regarding suicide and receive peer support [15]. According to Foundation Marketing, last year, Reddit found that, users average 21 billion screen views a month, meaning the users were spending considerable time engaging in content [16]. Reddit also showed that users that viewed Health related subreddits spent 15 times more time engaging than other users [16].

3.2 Data Collection

Both posts and comments were collected using the Pushshift application programming interface (API), which was created by the subreddit r/datasets mod team [17].

Data for this study was collected for the range of January 1st 2019 to December 31st 2020. However, data will continue to be collected to increase the dataset's size. As seen in Figures 1 and 2 below, certain months contain more posts than others. This trend seems to be consistent with the 2 years of collected data. Researchers have taken into account that this may skew or create biases. In the Active Machine Learning section, it will be discussed what was done to combat this. When collecting the data, there were many categories that the Pushshift API returned; however, only the categories

of id, author, date, score, number of comments, uniform resource locator (URL), title, and self-text were deemed relevant and kept. Once the data was extracted using the Pushshift API it was stored in a data frame that allows for easy import, export, and manipulation.

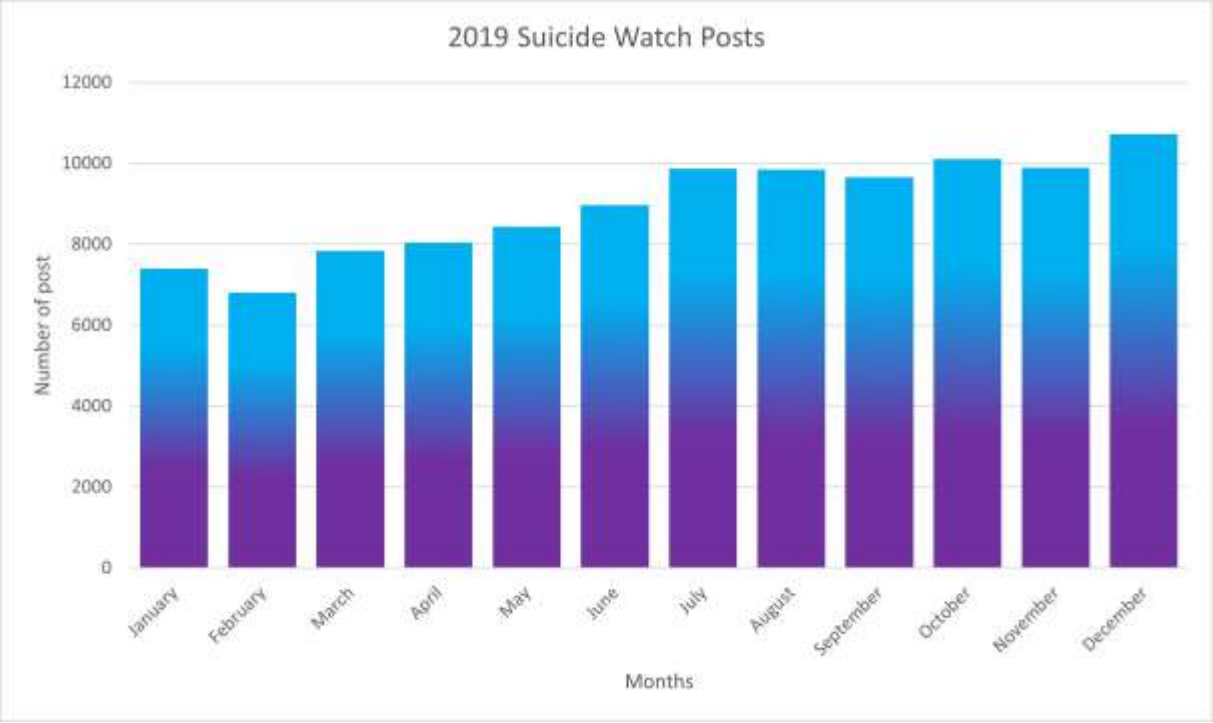

Figure 1: Graph of the number of posts per month in 2019

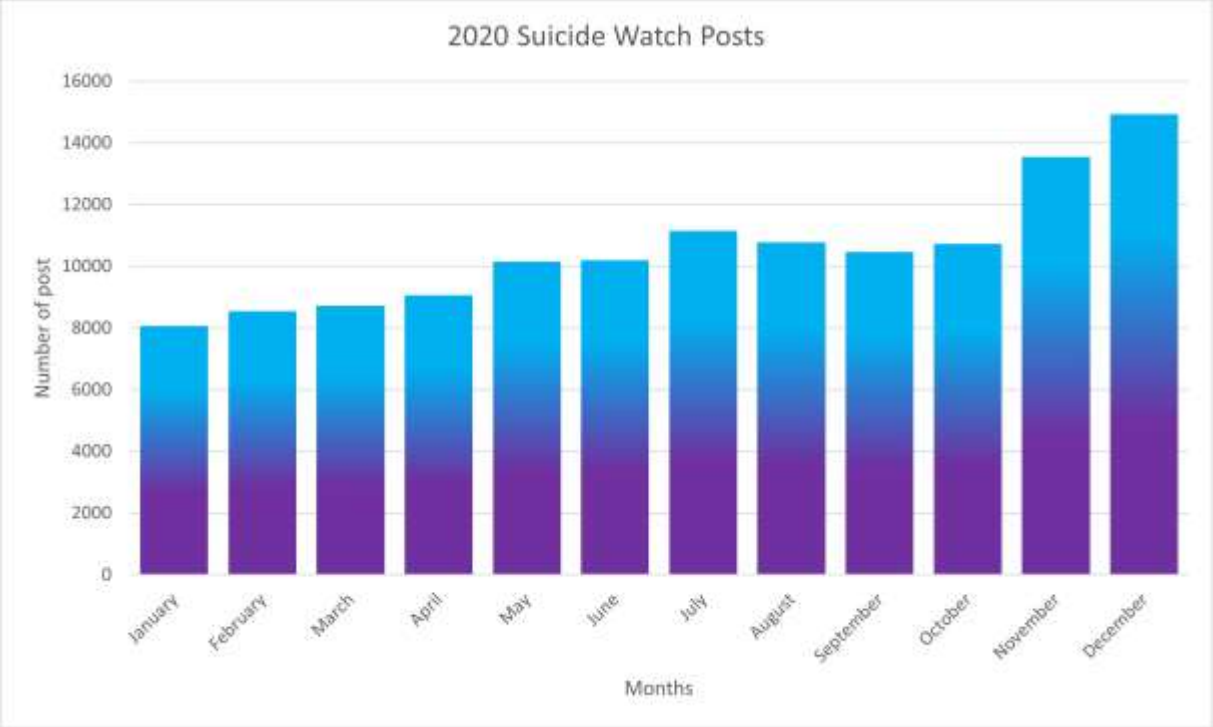

Figure 2: Graph of the number of posts per month in 2020

Index	ID	Author	Created UTC	Score	Number of comments	URL	Title	Selftext
0	bjcveg	sweets*uicide*	4/30/2019 23:57:42	2	1	URL to post	I hate life	Tell me life is going to get better. I have treatment resistant depression; no meds will work. I found relief after having 3 rounds of ketamine infusion therapy which took all my depression away for twoMonths. Now my life events make life not my worth living. I have the most supportive loving husband and beautiful healthy children. I have a great occupation and career but why isn't this good enough? I keep praying that this mental issues will go away but I can't stop crying
1	bjcvdq	Memegod697	4/30/2019 23:57:38	2	1	URL to post	I can't anymore.	Since elementary school I was always the kid that was full of potential. I liked the attention but now I wish it never happened. Fast forward to today, I fucked up. I made a mistake and now I'm a convicted felon at the age of 17 and im still waiting on a court date. I'm doing absolutely horrible in school. I have little to no friends and the one person who i thought cared about me is gone.
2	bjcuc5	throwaway2728 2929	4/30/2019 23:54:00	37	2	URL to post	I just want to fall asleep and never wake up.	Just little bit about me, I am a sophomore in high school whose had a pretty shit year. My parents got divorced because my Dad found out about her cheating with people from her work. My grades are shit because I'm so fucked in the head I can't read my own handwriting. I have to lose 10 pounds every week for wrestling one the only things I like about my life is wrestling and to be good you have to be as little weight as possible but everything else I love. The other things that's ok is video games it just lets me have a life that I would never have here. All the people I no in school make fun of for looks and overall stupidity. I just want to go peacefully so I don't have to deal with this shit anymore. If it doesn't get better soon I'm probably gonna take it into my own hands.
3	bjctsn	doodlefea	4/30/2019 23:52:14	2	4	URL to post	I reach for help and I miss everytime	I just can't seem to help myself, and I reach out for help but I don't get it. I feel like I can't do it anymore. I feel guilty for feeling this way because I love my bf so much and he is the only reason I haven't done it yet. But, sometimes I feel like its best for everyone if I was gone. Everyday I get closer.

Table 2: Example of data collected

4 Annotation

4.1 Manuel Coding

After collecting the data using the Pushshift API, a sample of 350 posts from November 2020 were initially taken to be manually coded. The manual coding was done by a collaboration of Taylor Griffin, Katherine Scoggins, and I under the supervision of Dr. Yiyi Yang. In "Exploring Cultural Influence on the Mediated Portrayal of Schizophrenia," the public and internalized stigmas of marks, group labeling, social exclusion, peril, and responsibility were derived using communication theory [18]. Along with the public and internalized stigma cues, challenge cues of optimism, hope, social inclusion, personification, and combat were also annotated. S. F. Scudder first proposed communication theory in the year 1980. It states that all living beings that exist on the planet communicate [19]. In this study, we are concerned about how people communicate through technology. While annotating public and internalized stigma cues, coders used the constant comparative method to derive the categories of emerging themes and the outcome variable. The emerging themes that were derived for this study were hopelessness, pessimism, exposure to abusive behavior, mental health issues, explicit expression of negative emotion, narcissism, bargaining, and remorse. The outcome variables that were created using the constant comparative method were self-harm/suicide attempts and ideation/planning. The constant comparative method is an inductive data coding process that categorizes and compares qualitative data for analysis purposes [20].

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}$$

(a) Cohen's Kappas equation

Value of Kappa	Level of Agreement	% of Data that is Reliable
0-.20	None	0-4%
.21-.39	Minimal	4-15%
.40-.59	Weak	15-35%
.60-.79	Moderate	35-63%
.80-.90	Strong	64-81%
Above .90	Almost Perfect	82-100%

(b) Table for determining data reliability

Figure 3: Cohen's kappa equation and table

4.2 Inter-Coder Reliability

Once the manual coders coded the small sample of data, inter-coder reliability was checked using Cohen's Kappa score. The Cohen's Kappa score was first introduced in Jacob Cohen's 1960 paper, "A Coefficient Of Agreement For Nominal Scales," which introduces a method to calculate the reliability of the data between coders [21]. Cohen's Kappa looks at the correctly coded categories compared against the errors along with the chance of random agreement to determine the reliability of the data. In this study, achieving a Kappa score of 75% is considered satisfactory by the researchers.

As seen in the graphic, some of the coded categories have achieved a score above the accepted value of 75%. These categories achieved a level that was either considered strong or almost perfect. However, it is also seen that some categories have fallen way below expectations, some even in the single digits. Another round of annotation was performed to see if the Kappa scores could be improved. Unfortunately, the scores did not improve significantly, and we were determined to only use two of the high-performing categories. These categories were Outcome – Self Harm/Suicide Attempts and Outcome Ideation/Planning, which are not stigmas. These categories were derived using the consent comparative method. This was decided so the research could move onto active machine learning to annotate the data. It also increased the training dataset for more sophisticated deep-learning algorithms. The researchers developed a methodology that was tested by using the two high-preforming categories and once the Kappa scores of the other categories reach a level of satisfaction, then active machine learning can be performed to add those categories to the training dataset.

Label	Percent	Label	Percent
Outcomes - Self-Harm/Suicide Attempts	85%	Outcomes - Ideation/Planning	94%
public stigma - marks	13%	public stigma - group labeling	16%
public stigma - social exclusion	66%	public stigma - peril	19%
public stigma - responsibility	33%	Internalized stigma - marks	12%
Internalized stigma - group labeling	5%	Internalized stigma - social exclusion	7%
Internalized stigma - peril	68%	Internalized stigma - responsibility	7%
Challenge cues - optimism	21%	Challenge cues - hope	13%
Challenge cues - social inclusion	71%	Challenge cues - personification	78%
Challenge cues - combat	32%	Emerging themes - hopelessness	59%
Emerging themes - pessimism	94%	ET - exposure to abusive behavior	92%
ET- mental health issues	92%	ET- Explicit Expression of Negative Emotion	87%
ET - Narcissism/Grandiosity	93%	ET - Bargaining	16%
ET- Remorse	15%	ET- Others	89%

Table 3: Cohen's Kappa scores table

5 Active Learning

5.1 Data Cleaning

After manually annotating the data to create the initial training dataset, the researchers used the query-by-committee approach in active machine learning to expand the training dataset. A query-by-committee approach is an active machine learning strategy, which reduces various disadvantages of uncertainty sampling by keeping several hypotheses at the same time and selecting queries where disagreement occurs [22]. Expanding the training dataset allowed for deep learning algorithms to be trained effectively. Before active machine learning could be performed, preparation of the data needed to be completed. The first step in data preparation is the partition of the title and body from a single data sample. Then the title and body are combined to create a single block of text. To continue pre-processing the data, the researchers replaced any number that appeared with the word "numbr". This was done to reduce sparsity and eliminate other vocabulary issues that may have arisen.

Another thing that was done to the dataset was to eliminate all white space before and after each sentence within every dataset sample. Also, data samples with multiple spaces between terms were replaced with a single space. The next pre-processing step was removing all punctuation from the data samples. Next, the researchers applied the .lower() method from the Python string library to make all of the words within each sample lowercase. Following that, a library and algorithm from nltk.corpus was imported and applied to remove all English stop words from the data samples [23]. A stop word is defined as a set of commonly used words within a given language [24]. It is important to remove these commonly used words so that the algorithm can focus on the important words within the samples when annotating. There are many predefined stopword lists; in this study, the list from nltk.corpus was used [23]. Also, from the nltk library, a port stemmer was used to remove stem words from the data sample [23]. Moving forward, the samples needed to be tokenized. For this task, WordPunctTokenize

was chosen [25]. WordPunctTokenizer was chosen because this tokenizer splits on white space and punctuation [25]. Researchers justified this tokenizer because they believed the separation of words allowed for the best machine-learning results to be achieved. This is shown by the figure below, which shows an acceptable level of accuracy. After tokenizing the words, the most frequently used words and two words were collected. From this, the top 30 two-word combinations were collected to create a lexicon. Shown below are the top 20 words and frequencies from the lexicon. These two-word combinations were chosen based on their frequency and relation to suicide. This lexicon was then applied to the dataset to combine words that would be included in the lexicon. This raised the accuracy of the model, as shown in Table 4 below. Finally, the training and testing labels were created and stored.

Words	Frequency	Words	Frequency
want,die	320	suicid,thought	36
get,better	261	self,harm	34
want,live	80	end,life	31
want,kill	70	get,wors	29
commit,suicide	64	die,want	28
get,help	51	tri,kill	27
want,end	48	mental,ill	26
live,life	41	suicid,attempt	25
need,help	40	pleas,help	25

Table 4: Top 20 words from lexicon

5.2 Machine Learning Algorithms

This study used a voting classifier to detect suicide attempts and suicide planning within a given social media post. With the voting classifier, machine learning algorithms were trained to predict separately then a poll was taken of their results to determine the overall prediction. Below is an explanation of those machine learning algorithms.

Machine learning is one of the many branches of artificial intelligence, and it was developed very early in the life of artificial intelligence. Machine learning focuses on using data to mimic how humans learn; it slowly improves its accuracy over time [26]. Arthur Samuel is credited with coining the term "machine learning" in his research based on the game of checkers [26]. Over the years, advances in technology have produced some powerful machine learning algorithms, with their applications including recommendation engines and self-driving cars [26]. Machine learning takes input data, which can be labeled or unlabeled, and estimates the outcome based on patterns in the data [26]. Machine learning algorithms fall into three primary categories: supervised, unsupervised, and semi-supervised [26]. Supervised learning is defined as using labeled data to train algorithms to classify data or predict outcomes [26]. Unsupervised learning is defined by its ability to discover hidden patterns or groupings with human interaction [26]. Semi-supervised learning is defined by its ability to combine both supervised and unsupervised learning by using a small dataset to guide the classification of a larger labeled dataset. Over the years, many algorithms have been developed within the machine learning framework. Those which are used in this study are included below.

Support Vector Machine - The Support Vector Machine (SVM) algorithm aims to find a hyperplane in an N-dimensional space, where N is the number of features that distinctly classify the data points [27]. Even though there are many possible hyperplanes to optimize the SVM, the algorithm maximizes the distance between the data points. The two types of kernels used during this study are linear and poly. The main difference between these kernels is how they define the hyperplane. Within the linear kernel, the hyperplane is drawn with a line y=x, while the poly kernel draws the hyperplane with a line $y = x^2$ [28].

Decision Tree – A decision tree uses a hierarchical structure consisting of a root node, branches, an internal node, and leaf nodes [29]. Each internal node is considered a decision node where the algorithm evaluates and then traverses a branch [29]. It continues to do this until a leaf, also known as a decision node, is reached [29]. Once a leaf

node is reached, this becomes the algorithm's prediction [29]. The main benefit of decision trees and why they were chosen for this study are that they need little data preparation and are very flexible.

Random Forest – Random Forest comprises a collection of decision trees, and each tree in the ensemble can be either correlated or uncorrelated [30]. Each decision tree is trained independently, and then the average or majority of those trees are taken to produce an overall result [30]. There are three main benefits of using a random forest algorithm and why it was chosen for this study. They reduce the risk of over-fitting, provide flexibility, and easily determine feature importance [30].

Stochastic Gradient Descent – This iterative method optimizes the gradient descent during each search [31]. The problem with gradient descent is that convergence to a local minimum might take extensive time and is not guaranteed [31]. Stochastic gradient descent seeks to determine the steepest descent to reduce the number of iterations [31]. Stochastic gradient descent is used in many machine learning algorithms, including support vector machines, logistic regression, and neural networks [31].

K-Nearest Neighbors – This is a non-parametric supervised learning classifier that uses proximity to make classifications or predictions based on the grouping of data points [32]. The algorithm first groups data points into classes, then when a prediction is needed, it plots the data points and then calculates the distances to the classes. The smallest distance is what point it is predicted as [32]. The advantages of K-Nearest Neighbor are that it is easy to implement, adapts to new data, and has few hyper-parameters [32]. Some disadvantages of K-Nearest Neighbor are it doesn't scale well, doesn't perform well with high-dimensional data, and is prone to overfitting [32].

Logistic Regression - Logistic regression is a statistical model that estimates the probability of an event occurring [33]. Logistic regression bounds are 0 to 1, and to get a prediction between these bounds, a logic transformation is applied to the odds [33]. There are three logistic regression models: binary, multinomial, and ordinal [33].

One thing to be wary of is that logistic regression is prone to over-fitting when there is a high number of predictor variables [33]. Some examples of logistic regression being utilized successfully are fraud detection, disease prediction, and churn prediction [33].

Voting Classifier – A voting classifier is a machine learning algorithm that trains numerous machine learning algorithms and makes a prediction based on the highest probability of the chosen class that has the majority [34]. There are two types of voting. Hard voting predicts the highest majority of votes, and soft voting predicts based on the average probability given to that class [34]. The greatest benefit of using a voting classifier is that it reduces the error present in a given model and allows for an average prediction.

5.3 Machine Learning Method

Now that the data preparation had been completed, active machine learning could take place. Five classifiers were chosen to build the voting classifiers: K Nearest Neighbors, Decision Tree, Random Forest, Logistics Regression, Stochastic Gradient Decent, and Support Vector Machine. A total of two classifiers were coded. Each of them dealt with only one of the chosen categories. Using the small, manually coded dataset, these models were trained and tested. Overall, they achieved approximately 50% accuracy. Since low accuracy was a concern, to move forward, a grid search was deployed to enact parameter tuning on the models. The five best models for both voting classifiers were chosen. The classifiers chosen for the "Outcomes suicide attempts" were Support Vector Machine (Poly), Support Vector Machine (Linear), Random Forest Classifier, Decision Tree Classifier, and Stochastic Gradient Decent. After grid search, the overall accuracy of the classifiers varied between 68% to 73% on the initial manually coded dataset. Each algorithm's sensitivity and specificity were taken along with judging the voting committee's overall accuracy. This was done to track how the algorithms reacted to more training data being introduced and small changes being made to the code. This data can be seen in Figure 4 and Tables 5, 6, and 7 below. To increase the amount of training, 20 data samples were taken every month from

2019 to 2020. This was done so that bias from a certain year or month could be minimized. The researchers then ran the samples through the active machine learning algorithms where the voting classifier would determine if there was any presence of suicide planning or suicide attempts. If greater than 75% of the classifiers agreed on the presence of suicide planning or suicide attempts, then a 1 would be coded. The same is true for the absence, except a 0 would be coded. If there was a disagreement between the classifiers, then an Oracle was prompted to determine the presence or lack of suicide planning and suicide attempts. Once the samples were annotated by the machine with help from the Oracle they were stored for a second validation by another Oracle. After the samples passed this second validation, they were then added to the training set. The algorithms were then retrained, and the parameters were tuned again. With the training dataset size increase, we see an increase in the metrics of the individual model and the voting classifier. We can see the transitions of each iteration in Tables 5, 6, and 7 below. Finally, deep learning can begin with the dataset increased to 1500 samples.

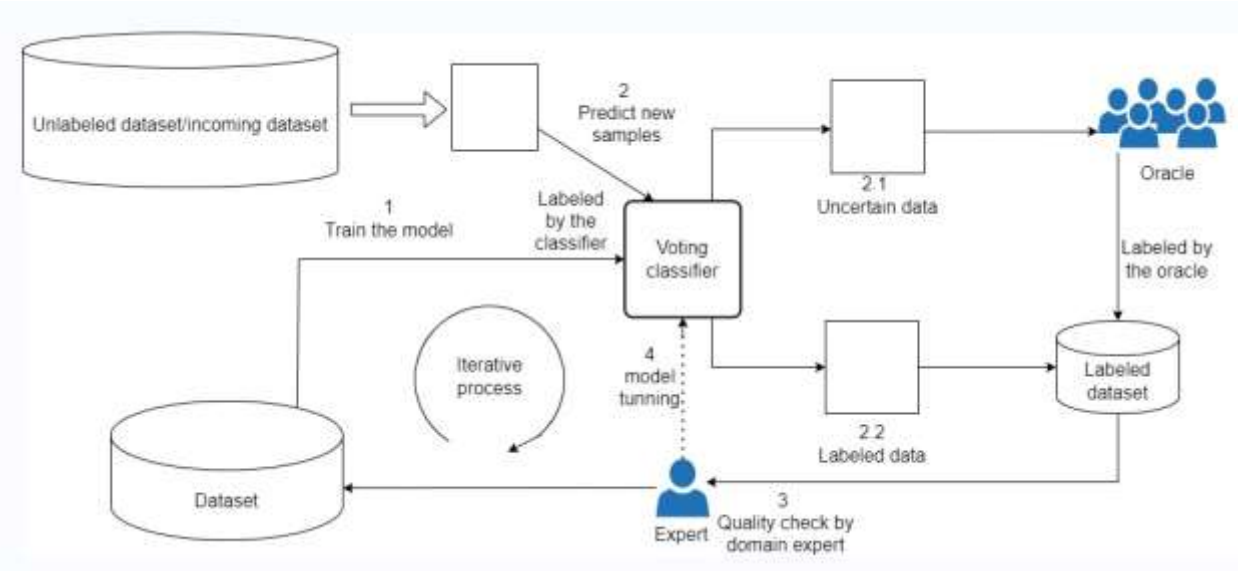

Figure 4: Model on active machine learning approach [49]

Run iteration (Committee)	Sensitivity	Specificity	Accuracy
Run 1 (Suicide Attempts)	36%	91%	77%
Run 3 (Suicide Attempts)	52%	94%	86%
Run 6 (Suicide Attempts)	60%	96%	93%
Run 1 (Suicide Planning)	85%	59%	75%
Run 3 (Suicide Planning)	89%	69%	82%
Run 6 (Suicide Planning)	93%	74%	90%

Table 5: Committee metrics table

Committee Member (Suicide Attempts)	Sensitivity (Run 1)	Specificity (Run 1)	Sensitivity (Run 3)	Specificity (Run 3)	Sensitivity (Run 6)	Specificity (Run 6)
SVC (Linear)	52%	83%	63%	91%	96%	97%
SVC (Poly)	32%	89%	55%	94%	98%	94%
Random Forest	27%	92%	40%	93%	97%	88%
Decision Tree	52%	80%	54%	85%	89%	91%
SGD	51%	76%	56%	87%	90%	85%

Table 6: Attempts Member metrics table

Committee Member (Suicide Planning)	Sensitivity (Run 1)	Specificity (Run 1)	Sensitivity (Run 3)	Specificity (Run 3)	Sensitivity (Run 6)	Specificity (Run 6)
SGD (Linear)	90%	46%	91%	61%	60%	99%
SVC (Poly)	77%	61%	88%	70%	99%	97%
KNN	62%	64%	40%	87%	96%	91%
Logistic	85%	54%	85%	77%	74%	79%
Decision Tree	70%	59%	80%	62%	88%	98%

Table 7: Planning Member metrics table

6 Deep Learning

6.1 Baseline Machine learning algorithm

Naive Bayes – Naive Bayes is a collection of classifiers that predict the probability based on Bayes' Theorem [35]. In order to have Naive Bayes be a fast and easy algorithm but still achieve a high-level accuracy, an assumption that feature values are independent of the given label [36]. There are three types of Naive Bayes: Gaussian, Multinomial, and Bernoulli [35]. In Gaussian Naive Bayes, each feature is assumed to be distributed according to a Gaussian distribution [35]. In Multinomial Naive Bayes, each feature is assumed to be distributed according to a multinomial distribution [35]. Finally, in Bernoullio, features are independent boolean describing inputs [35].

Temporal Convolutional Networks – A Temporal Convolutional Network consists of two steps. First, it computes low-level features using a Convolutional Neural Network that encodes spatial-temporal information [37]. After this is accomplished, these low-level features are input into a classifier that captures high-level temporal information using a Recurrent Neural Network [37]. This approach allows the capture of two levels of information in a hierarchy model [37]. One main feature of Temporal Convolutional Networks is that they can take a series of any length and output it as the same length [37]. Finally, Temporal Convolutional Networks have been used successfully recently in weather prediction as they outperformed long short-term memory networks [37].

6.2 Naive Bayes Method

A Naive Bayes was coded to provide a baseline for the Temporal Convolutional Neural Network discussed in a later section. The data preparation discussed in an early section used for machine learning remained the same for the Naive Bayes. To branch off from the previous machine learning code, a tfidfvectorizer (that converts raw documents to a matrix of TF-IDF features) was imported from the scikit-learn module "feature extraction" to continue prepping the data for the Naive

Bayes [38]. Also, the Gaussian Naive Bayes was imported from the scikit-learn module Naive Bayes [39].

Once all the modules needed for prepping and training had been imported, the tfidfvectorizer dot fit transform was performed on the processed data. The labels for the data were also stored in a variable. After transforming the data using the tfidfvectorizer, the train-test split from the scikit-learn module model selection was imported and applied [40]. During application, the Pandas method "To Dense" was applied to the data to convert the data into a dense array [41]. This was done because the Naive Bayes needed a dense array as an input, and the data was not stored in that form. The test size parameter was defined as 40%. This was determined as a good baseline split of the data. After the splitting of the data, training of the Naive Bayes occurred.

After we had trained the Naive Bayes model, it was then used to predict the testing set of data. The model was then evaluated by importing metrics from the scikit-learn package [42]. The model's accuracy was found to be 66% for predicting attempts and 79% for planning. Next, a confusion matrix was produced, knowing that false positives were acceptable and false negatives were not. As shown below, for attempts, we can see that we have 373 true positives and 34 true negatives in Table 8. What is concerning about this model is the high number of false negatives. This means that the model failed to recognize a data sample that contained suicide attempts within it. For attempts, it was calculated that the sensitivity was 71% and that the specificity was 36%. Looking at the planning matrix, we see that we have 77 true positives and 407 true negatives. Compared to the attempts model, we see only 95 false negatives. This was better but still very high. For planning, it was calculated that the sensitivity was 45% and the specificity 91%. To sum up, the Naive Bayes provided a baseline for the temporal neural network discussed in later sections and has performed worse than the machine learning algorithms. The Naive

Bayes model also had a high number of false negatives. It is believed that parameter tuning and retraining of the Naive Bayes could obtain results greater than or equal to the machine learning algorithms.

actual value		Prediction outcome for Attempts p	Prediction outcome for Attempts n	Total
	p′	373	149	522
	n′	61	34	95
	Total	434	183	

actual value		Prediction outcome for planning p	Prediction outcome for planning n	Total
	p′	77	95	172
	n′	38	407	445
	Total	115	502	

Table 8: Confusion Matrix for Naive Bayes Model

6.3 Temporal Neural Network (TCN) Method

Starting with the prepared data from the earlier section, we first define a function "max_length" to compute the max length of a given sequence. This function will be used later on in order to create the TCN model. Next, we load the GloVe word vector dictionary [43]. GloVe is an unsupervised learning algorithm for obtaining vectors related to words [43]. The training that occurs is on aggregated global word-word co-occurrence statistics from a corpus [43]. The resulting representation is showcased in a linear substructure of the word vector space [43]. After loading the GloVe word vector dictionary, we then define a function, "training_words_in_word2vector", to count the number of words present in the pre-trained word vector. Next, the training content is tokenized using the tokenizing from the nltk library and setting oov_token (out of vocabulary) equal to unknown (UNK) [44]. Following tokenizing, we find the number of words present in the training vocabulary by using the training_words_in_word2vector function. There were 8307 words present from the 8522 words in the training vocabulary. Continuing, we define a function, get_mean_vector, that will return an embedded dictionary's mean and standard deviation. These computations will be used in the pre-trained embedding matrix discussed next. A pre-trained embedding matrix was defined using the word vector map, word index, mean, and standard deviation. We set the vocabulary size to the length of the word index plus one. Next, we defined the dimensionality and initialized the matrix using the np.random.normal function passes in the mean, standard deviation, vocab size, and dimension [45]. The final step to complete the pre-trained embedding matrix was to set each row index as the word vector representing the index. We finally reached the point where we could define our Temporal Convolutional Network model.

This model's layers are sequential and connected to the previous layer. The first layer of our model is an input layer with an output shape of (None, 100). The data was then passed into the embedding layer, which had the weights set to the pre-trained embedding

matrix defined above, and the weights were no longer trainable. This layer had an output shape of (None, 100, 300), and 300,000 parameters. The next layer is the spatial dropout, which was set to 0.1 and had the same output shape as the previous layer but no parameters. We then come to the first of two TCN layers; they both had dilations of [1, 2, 4], return_sequences equal to true, and the activation function was set to ReLU (Rectified Linear Unit). Where the two TCN layers differ is in the number of filters. The first layer has 128 filters, while the second layer only has 64. The first TCN layer had an output shape of (None, 100, 128) and 400,256 parameters, while the second layer had an output shape of (None, 100, 64) and 94,656 parameters. The next two layers, global average pooling, and global max pooling, with output shape of (None, 64), are connected to the 2nd TCN layer. We find each feature map's average and max output from the previous layer in these layers. These are then passed into the concatenate layer. In this layer, which has an output shape of (None, 128), we concatenate the max and average global pooling to pass it into our next layer. The next layer is a dense layer, with an output of 16, the activation as relu, and 2064 Parameters. Our second to the last layer was a dropout layer with a rate equal to 0.1 and an output shape of (None, 16). The final layer of our model was another dense layer that had an output shape of (None, 1) and 17 parameters, which was the final prediction. The model was then compiled using binary_crossentropy loss function, Adam optimization, and metrics equal to accuracy [46]. Below in Figure 5 is an image of the TCN model created.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 100)]	0	[]
embedding (Embedding)	(None, 100, 300)	300000	['input_1[0][0]']
spatial_dropout1d (SpatialDropout1D)	(None, 100, 300)	0	['embedding[0][0]']
tcn1 (TCN)	(None, 100, 128)	400256	['spatial_dropout1d[0][0]']
tcn2 (TCN)	(None, 100, 64)	94656	['tcn1[0][0]']
global_average_pooling1d (GlobalAveragePooling1D)	(None, 64)	0	['tcn2[0][0]']
global_max_pooling1d (GlobalMaxPooling1D)	(None, 64)	0	['tcn2[0][0]']
concatenate (Concatenate)	(None, 128)	0	['global_average_pooling1d[0][0]', 'global_max_pooling1d[0][0]']
dense (Dense)	(None, 16)	2064	['concatenate[0][0]']
dropout (Dropout)	(None, 16)	0	['dense[0][0]']
dense_1 (Dense)	(None, 1)	17	['dropout[0][0]']

Total params: 796,993
Trainable params: 496,993
Non-trainable params: 300,000

Figure 5: Model on TCN approach

In order to prevent overfitting and optimize training, the EarlyStoping function from the keras callbacks library was used [46]. In this function, we monitored the validation accuracy, with the minimum delta equal to 0, patience equal to 10, verbose equal to 2, and mode equal to auto, and we restored the best weights when early stopping occurred. Now that the model had been defined, training on our dataset was time to occur. For cross-validation, KFold equal to 5 was chosen because it was determined to be optimal for achieving the desired results in the allotted time [47]. It should also be noted that shuffle was equal to true. The training data was then split into training and testing groups using the split function of KFold [48]. Then we turn the labels for the data into numpy arrays so that they can be used with the model. We clean and tokenize the training and testing data using the tokenizer described above. Afterward, tokenized padding occurs to ensure all

sequences have the same size. Now that the data was tokenized and padded, we initialized the pre-trained embedding matrix and the TCN model. Finally, we trained the model on the data with a batch size of 50, epochs 100, verbose 1, and the callback equal to the EarlyStopping as defined earlier. To evaluate the model, the accuracy of prediction on the test data was taken; Tables 9 and 10 are the training results.

Run	Activation	Filters	Average Accuracy
0	relu	1	89.81
1	relu	2	89.55
2	relu	3	89.94
3	relu	4	89.68
4	relu	5	89.61
5	relu	6	90.07
6	relu	7	89.74
7	relu	8	89.49

Table 9: Training results of the TCN model for Attempts

Run	Activation	Filters	Average Accuracy
0	relu	1	86.68
1	relu	2	84.41
2	relu	3	87.98
3	relu	4	85.06
4	relu	5	86.36
5	relu	6	85.38
6	relu	7	86.68
7	relu	8	85.38

Table 10: Training results of the TCN model for planning

Once we had the models trained, predictions on the dataset could occur. Unfortunately, the model makes a prediction between 0 and 1, and the researcher's parameters are either 0 or 1. To overcome this problem, a threshold was put in place in order to categorize the predictions into either 0 or 1. This was based on whether they were above or below the threshold. The next problem revolved around which threshold to use. To solve this problem, more testing occurred to determine the best threshold. Each threshold was tested from 0.0 to 1.0 using an F1 score as the metric to measure the threshold viability. As a

baseline to evaluate if the found threshold was truly optimal, the researchers also tested with a .50 threshold. For attempts, the best threshold determined by the algorithm was .36. For planning, the best threshold found was 0.68. After finding the optimal threshold, the researchers then used the trained models to make a prediction and categorized it as 1 or 0, depending on whether it was above or below the baseline threshold and the found threshold. After predictions were made, a confusion matrix was taken to compare the thresholds and the baseline Naive Bayes model. Below are the confusion matrices for each model in Table 11.

The results of this study have proven to be very interesting and both expected and unexpected. Looking at the dataset, we can see a trend in the number of monthly posts being higher during the year's later months. During the manual coding portion of this study, only 8 of the 26 were deemed the acceptable level of moderate using the Cohen Kappa score. This study used the two categories' attempts and planning to make predictions. The data cleaning was very industry standard, except for using the WordPunctTokenizer for the voting classifier and the Glove word vector dictionary. It should also be noted that a lexicon of bi-words was developed to raise the accuracy of the models. Next, we can look at the thresholds used for predicting data in the TCN models. As seen below in Table 11, the confusion matrices for attempts, the threshold the algorithm found, outperformed the .50 threshold determined by researchers. However, this was not true regarding the model used for planning. There we can see that the .50 threshold set by researchers outperformed the algorithm-found threshold. Looking at all three models, we can see that the voting classifier had the highest accuracy at 93% for attempts and 90% for planning. The second best in terms of accuracy is the TCN model, which resulted in 89% accuracy for attempts and 90% for planning. Performing the lowest out of the three models was the Naive Bayes, with a 66% accuracy for attempts and 79% accuracy for planning. It is not unexpected to have these results as the voting classifier had the most human intervention, and the TCN model is often considered a more robust algorithm than Naive Bayes. It should also be noted that more time was spent on coding the TCN over the Naive Bayes, and therefore it should

be expected to outperform it. Even though accuracy is a great metric for comparing algorithms, we dove deeper into the comparison between the TCN model and Naive Bayes confusion matrices. As seen in Table 8, the Naive Bayes model had the truest positives and truest negatives over the TCN model. However, the Naive Bayes had more true negatives and true positives. It also had more false negatives, which was the metric that the researcher focused on after the true positives and true negatives. As seen in Table 8, the Naive Bayes had over double the false negatives in some cases. The reason false negatives are so important is the researcher would rather mislabel the data as positive rather than negative. When it comes to suicide, it is better to error on caution. To summarize all the results, the study showed that the voting classifier performed the highest when accuracy metrics were used and had the most human interaction. Following the voting classifier, the TCN model outperformed the Naive Bayes model in accuracy and confusion matrices.

Prediction outcome for Attempt(.36 Threshold)

actual value	p	n	Total
p′	253	15	268
n′	19	21	40
Total	272	36	

Prediction outcome for Attempt(.50 Threshold)

actual value	p	n	Total
p′	264	4	268
n′	24	16	40
Total	288	20	

Prediction outcome for Planning(.68 Threshold)

actual value	p	n	Total
p′	52	28	80
n′	21	207	228
Total	73	235	

Prediction outcome for Planning(.50 Threshold)

actual value	p	n	Total
p′	43	37	80
n′	8	220	228
Total	51	257	

Table 11: Confusion Matrices for TCN Models

7 Conclusion

In conclusion, artificial intelligence methods, machine learning, and deep learning can perform tasks humans can do in less time and achieve similar, if not better, results. However, not every artificial intelligence algorithm is suitable for a particular problem, as demonstrated by this research. A combination of data preparation, artificial intelligence algorithms, and optimization is needed to achieve the desired results. Despite the TCN not being optimized fully, it was still able to achieve results similar to the voting classifier and human annotators. It was also able to outperform the Naive Bayes model. This research has shown that while stigma classification can be a complex problem, a deep-learning TCN model is a viable option to employ.

8 Future Work

Based upon the results of this research, other questions have arisen that could provide the foundation for future research. Firstly, only two consecutive years of data were collected. Adding more years of data to remove any short-term trends could prove worthwhile. As discussed in the conclusion section, the non-optimized TCN Model performed almost as well as the machine learning algorithms, outperforming the Naive Bayes model. Exploring optimizing the TCN model to achieve results greater than the machine learning algorithms could be worthwhile. It also could be noted that running the algorithms on a more powerful machine could improve performance. Another avenue for future work would be to apply the research method and TCN model to the other dimensions of stigma mentioned in the annotation section. This could lead to a model that could annotate and predict all the dimensions mentioned in the annotation section. Finally, exploring other deep learning algorithms could provide results greater than those achieved in this study. Do remember that the same measuring of the algorithm's performance must be used to provide significant evidence.

References

[1] "Fast facts," May 2021. [Online]. Available: https://www.cdc.gov/suicide/facts/index.html

[2] S. Kučukalić and A. Kučukalić, "Stigma and suicide," *Stigma and Suicide.*, 2017. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/29283986/

[3] "stigmatization." [Online]. Available: https://dictionary.cambridge.org/us/dictionary/english/stigmatization

[4] "Better ways to prevent suicide." [Online]. Available: https://www.apa.org/monitor/2019/07-08/cover-prevent-suicide

[5] "Demographics of social media users and adoption in the united states," Apr 2021. [Online]. Available: https://www.pewresearch.org/internet/fact-sheet/social-media/

[6] "Oracle definition amp; meaning." [Online]. Available: https://www.merriam-webster.com/dictionary/Oracle

[7] L. Schweitzer, "Planning and social media: A case study of public transit and stigma on twitter," *Journal of the American Planning Association*, vol. 80, no. 3, p. 218–238, 2014.

[8] Q. Cheng, T. M. Li, C.-L. Kwok, T. Zhu, and P. S. Yip, "Assessing suicide risk and emotional distress in chinese social media: A text mining and machine learning study," *Journal of Medical Internet Research*, vol. 19, no. 7, 2017.

[9] N. Oscar, P. A. Fox, R. Croucher, R. Wernick, J. Keune, and K. Hooker, "Machine learning, sentiment analysis, and tweets: An examination of alzheimer's disease stigma on twitter," *The Journals of Gerontology: Series B*, vol. 72, no. 5, p. 742–751, 2017.

[10] J. Song, T. M. Song, D.-C. Seo, and J. H. Jin, "Data mining of web-based documents on social networking sites that included suicide-related words among korean adolescents," *Journal of Adolescent Health*, vol. 59, no. 6, p. 668–673, 2016.

[11] A. L. Nobles, J. J. Glenn, K. Kowsari, B. A. Teachman, and L. E. Barnes, "Identification of imminent suicide risk among young adults using text messages," *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018.

[12] F. Tokmic, M. Hadzikadic, J. R. Cook, and O. V. Tcheremissine, "Development of a behavioral health stigma measure and application of machine learning for classification," Jun 2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6040723/

[13] D. CURRY, "Reddit revenue and usage statistics (2022)," Jan 2022. [Online]. Available: https://www.businessofapps.com/data/reddit-statistics/

[14] "The history of reddit," Aug 2020. [Online]. Available: https://www.honorsociety.org/articles/history-reddit

[15] R. Reader, "Reddit will now automatically connect potentially suicidal users with a hotline," Mar 2020. [Online]. Available: https://www.fastcompany.com/90472072/reddit-will-now-automatically-connect-potentially-suicidal-users-with-a-hotline

[16] F. Team, "Reddit statistics for 2021: Eye-opening usage amp; traffic data," Jan 2022. [Online]. Available: https://foundationinc.co/lab/reddit-statistics/

[17] Pushshift, "pushshift/api." [Online]. Available: https://github.com/pushshift/api

[18] Y. Yang and S. Parrott, "Schizophrenia in chinese and u.s. online news media: Exploring cultural influence on the mediated portrayal of schizophrenia," *Health Communication*, vol. 33, no. 5, p. 553–561, 2017.

[19] "Msg management study guide." [Online]. Available: https://www.managementstudyguide.com/communication-theory.htm

[20] J. Olson, C. Mcallister, L. Grinnell, K. G. Walters, and F. Appunn, "Applying constant comparative method with multiple investigators and inter-coder reliability," *The Qualitative Report*, 2016.

[21] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, p. 37–46, Apr 1960.

[22] T. Danka, "Query by committee¶," 2018. [Online]. Available: https://modal-python.readthedocs.io/en/latest/content/examples/query_by_committee.html#:~:text=Query%20by%20committee%20is%20another,the%20sight%20of%20the%20estimator.

[23] "nltk.corpus package," Mar 2022. [Online]. Available: https://www.nltk.org/api/nltk.corpus.html

[24] K. Ganesan, "What are stop words?" Jul 2020. [Online]. Available: https://kavita-ganesan.com/what-are-stop-words/

[25] "nltk.tokenize package," Mar 2022. [Online]. Available: https://www.nltk.org/api/nltk.tokenize.html

[26] I. C. Education, "What is machine learning?" Jul 2020. [Online]. Available: https://www.ibm.com/cloud/learn/machine-learning#toc-how-machin-NoVMSZI_

[27] R. Gandhi, "Support vector machine - introduction to machine learning algorithms," Jul 2018. [Online]. Available: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[28] M. Oleszak, "Svm kernels: What do they do?" Sep 2021. [Online]. Available: https://towardsdatascience.com/svm-kernels-what-do-they-actually-do-56ce36f4f7b8

[29] N. Chauhan, "Decision tree algorithm, explained," Feb 2022. [Online]. Available: https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html

[30] I. C. Education, "What is random forest?" Dec 2022. [Online]. Available: https://www.ibm.com/cloud/learn/random-forest

[31] J. Price, A. Wong, T. Yuan, J. Mathews, and T. Olorunniwo, "Stochastic gradient descent," Dec 2020. [Online]. Available: https://optimization.cbe.cornell.edu/index.php?title=Stochastic_gradient_descent

[32] I. IBM, "What is the k-nearest neighbors algorithm?" [Online]. Available: https://www.ibm.com/topics/knn#:~:text=The20k2Dnearest20neighbors20algorithm2C20also20known20as20KNN20or,of20an20individual20data20point.

[33] ——, "What is logistic regression?" [Online]. Available: https://www.ibm.com/topics/logistic-regression#:~:text=Logistic20regression20estimates20the20probability,bounded20between20020and201.

[34] A. Kuwar, "Ml: Voting classifier using sklearn," Nov 2019. [Online]. Available: https://www.geeksforgeeks.org/ml-voting-classifier-using-sklearn/

[35] N. Kumar, "Naive Bayes classifiers," Feb 2022. [Online]. Available: https://www.geeksforgeeks.org/naive-bayes-classifiers/

[36] C. Cornell, "Bayes classifier and naive bayes," Jul 2018. [Online]. Available: https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote05.html

[37] B. Or, "Temporal convolutional networks, the next revolution for time- series?" Feb 2022. [Online]. Available: https://towardsdatascience.com/temporal-convolutional-networks-the-next-revolution-for-time-series-8990af826567

[38] "6.2. feature extraction." [Online]. Available: https://scikit-learn.org/stable/modules/feature_extraction.html

[39] "Sklearn.naive_bayes.gaussiannb." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html

[40] "Sklearn.model_selection.train_test_split." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

[41] "Pandas.dataframe.sparse.to_dense." [Online]. Available: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.sparse.to_dense.html

[42] "3.3. metrics and scoring: Quantifying the quality of predictions." [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html

[43] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," 2014. [Online]. Available: https://nlp.stanford.edu/projects/glove/

[44] "nltk.tokenize package," Mar 2022. [Online]. Available:https://www.nltk.org/api/nltk.tokenize.html

[45] "Numpy.random.random." [Online]. Available: https://numpy.org/doc/stable/reference/random/generated/numpy.random.random.html

[46] "Tf.keras.model nbsp;: nbsp; tensorflow v2.10.0," Oct 2022. [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/Model

[47] "3.1. cross-validation: Evaluating estimator performance." [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html

[48] "Sklearn.model_selection.kfold." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html

[49] Tianrui Liu et al. Enriching an Online Suicidal Dataset with Active Machine Learning. 2022.