A Natural Language Processing Analysis of Reddit as a Social Media Platform for Discussions on Mental Health

Joseph Tomasello and Daehan Kwak (Faculty Advisor)
School of Computer Science and Technology, Kean University, Union, NJ 07083

CPS-4961 Section 02
Spring 2023

Table of Contents

Abstract

Over the last decade, mental health has emerged as a topic of intense interest to the general populace, having finally shed much of stigmatization that kept it out of public discourse for so long. In fact, discussions on the subject are cropping up seemingly everywhere, particularly online, with many popular social-oriented platforms now featuring spaces set up specifically for users to offer up any relevant thoughts, opinions, advice, or even first-hand experience they may have with the subject. Some of the most heavily trafficked examples over the past few years can be found on Reddit.com, an American social news discussion platform divided into individual forums categorized by content, referred to as either “communities” or “subreddits” on the site. The three largest subreddits within the domain of mental health are the aptly-named r/mentalhealth, r/depression, and r/Anxiety, boasting over 1.8 million subscribers between them.

Introduction

Across each of the three target subreddits, hundreds of users congregate daily to submit posts and discuss the topic of mental health from all conceivable angles. Through extracting this data from Reddit’s leading mental health-centric communities and analyzing any apparent trends, information on how the subject is being examined by the site’s userbase can be obtained.

Objective

As an objective, this project set out to aggregate four years’ worth of post-based textual data from the three subreddits cited above over a collection period spanning from March 11, 2018 to March 11, 2022. Then, using natural language processing

methodologies, the resulting dataset could be anatomized both in terms of a sentiment analysis and topic modeling. In taking this approach, any trends extending across all three subreddits, along with each one individually, were sought to provide a sense of how mental health is most often approached and viewed by the site's contributing userbase, along with precisely what topics commonly lie at the forefront of mental health discussion on the site. Functionally, this involved honing in on both the body text of the dataset's posts, as well as the ways in which posts are subcategorized within each community utilizing the site's tagging system.

On the sentiment analysis end, the proportion of posts that were found to be primarily negative in tone versus the amount found to be chiefly positive was of particular interest in order to provide a sense of how Reddit's users seem to be grappling with the subject of mental health, broadly speaking. On a more granular level, specific emotions expressed were also desired in order to expand on these findings, further illustrating what feelings have been acting as the driving force for the bulk of the content across the targeted communities.

On the topic modeling end, a key area of focus was the count of each unique significant term used across the post body text of the dataset. With this information, the true semantic structure of the site's posts could be further uncovered to reveal the most widely discussed issues surrounding mental health on Reddit. Additionally, tag data combed from the three target subreddits, referred to as "flairs" on the site, could also be modeled to typify post content, examples of which include "Advice Needed", "Helpful Tips", and "Venting", among others.

Methods / Implementation

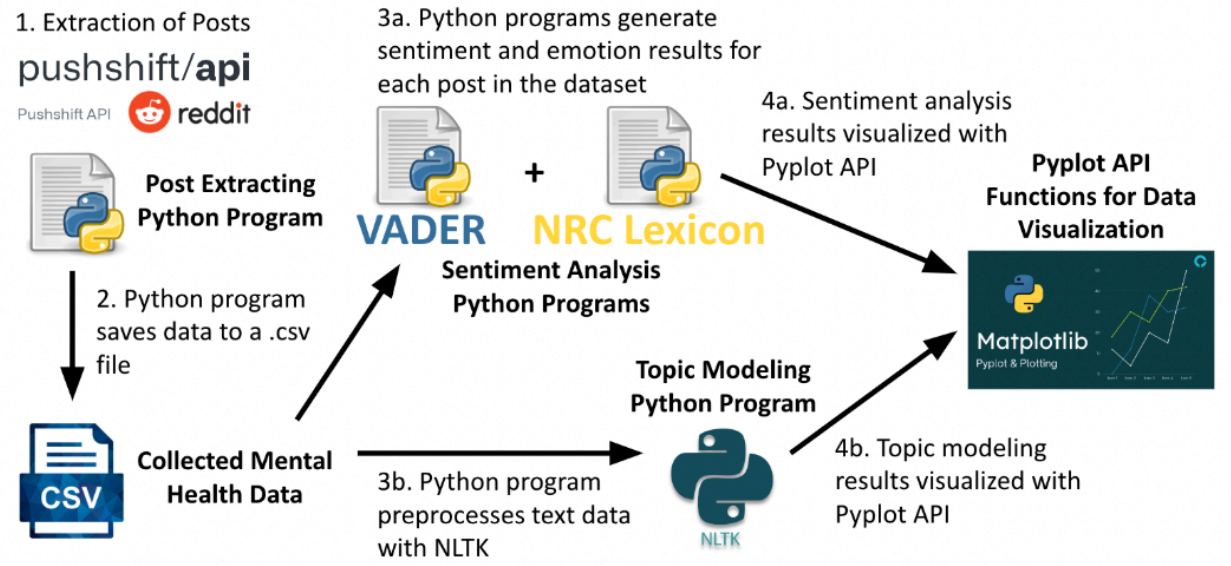


Figure 1 - Overall Approach to Extracting Posts and Subsequent Analysis

Pushshift API

```
from psaw import PushshiftAPI                    #Importing wrapper library for reddit(Pushshift)
import datetime as dt                            #Importing library for date management
import pandas as pd                              #Importing library for data manipulation in python
import matplotlib.pyplot as plt                  #Importing library for creating interactive visualizations in Python
from pprint import pprint                        #Importing for displaying lists in the "pretty" way (Not required)

pd.set_option("display.max_columns", None)       #Configuration for pandas to show all columns on dataframe
api = PushshiftAPI()                             #We create an object of the API
```

Figure 2 - Pushshift API and Associated Imports in Python

```python
"""FOR POSTS"""
def data_prep_posts(subreddit, start_time, end_time, filters, limit):
    if(len(filters) == 0):
        filters = ['id', 'author', 'created_utc', 'domain', 'url', 'title',
                   'num_comments', 'selftext', 'link_flair_richtext']
                   #We set by default some columns that will be useful for data analysis

    posts = list(api.search_submissions(
        subreddit=subreddit,                          #We set the subreddit we want to audit
        after=start_time,                             #Start date
        before=end_time,                              #End date
        filter=filters,                               #Column names we want to get from reddit
        limit=limit))                                 #Max number of posts we wanto to recieve

    return pd.DataFrame(posts)                        #Return dataframe for analysis
```

Figure 3 - Python Code for Post Data Collection Using the Pushshift API

In order to audit the target subreddits, access to Reddit and its inventory of posts was facilitated through use of the Pushshift API, in combination with Python, and then manipulated and saved using the DataFrame data structure native to the Pandas software library (Figure 2 and Figure 3). This resulted in a dataset consisting of approximately 1.25 million posts sourced from r/mentalhealth, r/depression, and r/Anxiety.

Sentiment Analysis (Methods / Implementation)

The sentiment analysis portion of the project was accomplished using the Python-based VADER (Valence Aware Dictionary and sEntiment Reasoner) and NRC Emotion Lexicon libraries.

```python
# This function generates a sentiment polarity figure and assigns a value of "Negative",
# "Neutral", or "Postive" to each collected post
def vader_analysis(dataframe):

    # Adds a new column to the dataframe to store a set of sentiment polarity figures
    # for the CleanBodyText of each collected post
    dataframe["vader_polarity"] = dataframe['clean_body_text'].apply(
        lambda x: sid.polarity_scores( str(x) ) )

    # Adds a new column to the dataframe to store the compound score from the polarity
    # figures
    dataframe['compound'] = dataframe['vader_polarity'].apply(
        lambda score_dict: score_dict['compound'] )

    # Adds a new column to the dataframe to store either a "Negative", "Neutral", or
    # "Postive" designation
    dataframe['vader_sentiment'] = dataframe['compound'].apply(
        lambda x: translate_vader_sentiment(x) )
```

Figure 4 - VADER Python Code for Sentiment Analysis

```python
# This helper function translates the numerical compound polarity score to a text score
# reading either "Negative", "Postive", or "Neutral"
def translate_vader_sentiment(compound):

    if compound <= -0.05:
        return "Negative"
    elif compound >= 0.05:
        return "Positive"
    else:
        return "Neutral"
```

Figure 5 - VADER Python Code for Score Translation

The VADER library is specifically attuned to the identification of sentiments expressed in social media and was applied to the body text of each collected post (Figure 4). The result was a numerical score between -1 and 1, which is indicative of an overall designation of either “Negative”, “Positive”, or “Neutral” (Figure 5). These results were then saved via new columns added to the working DataFrame.

```python
# This function generates a set of raw emotion/sentiment scores for each post
def nrc_lexicon_analysis(dataframe):

    # Adds a new column to the dataframe to store a set of emotion/sentiment scores
    # for the clean_body_text of each collected post
    dataframe["nrc_emotion_scores"] = dataframe['clean_body_text'].apply(
        lambda x: NRCLex( str(x) ).raw_emotion_scores )

    return dataframe
```

Figure 6 - NRC Lexicon Python Code for Sentiment Analysis

The NRC Emotion Lexicon library extends the basic sentiment analysis premise to also include a range of eight, more pinpointed emotions: anger, fear, anticipation, trust, surprise, sadness, joy, and disgust. To do this, the above code was used to generate a raw numerical score for each of the eight identifiable emotions found across each post (Figure 6). Likewise, these results were also added to the working sentiment analysis DataFrame for the purpose of visualizing later.

Topic Model (Methods / Implementation)

```python
import nltk
wn = nltk.WordNetLemmatizer()
ps = nltk.PorterStemmer()
stopword = nltk.corpus.stopwords.words('english')
```

Figure 7 - NLTK Imports in Python

```python
# This helper function removes all characters and text not relevant to performing a topic
# analysis (i.e., special characters, punctuation, and URLs)
def clean_text_TA(text):

    text = text.replace('\n', ' ')           # Removes any newline characters
    text = text.replace('\r\r', ' ')         # Removes any carriage return characters
    text = text.replace('&amp;#x200B;', ' ')  # Removes any zero-width space characters
    text = re.sub(r'http\S+', '', text)      # Removes any URLs beginning with http
    text = re.sub(r'www\S+', '', text)       # Removes any URLs beginning with www
    text = text.replace('_', ' ')            # Removes any _ characters
    text = text.replace('--', ' ')           # Removes any -- characters
    text = text.replace('-', ' ')            # Removes any - characters
    text = text.replace(',', ' ')            # Removes any , characters (Example: Also,I would... --> Also I would...)
    text = text.replace('"', ' ')            # Removes any " characters (Example: An "easy"task --> An easy task)
    text = text.replace('/', ' ')            # Removes any / characters (Example: small/medium/large --> small medium large)
    text = text.replace('(', ' ')            # Removes any ( characters (Example: His(Kevin's)car --> His Kevin's)car)
    text = text.replace(')', ' ')            # Removes any ) characters (Example: His Kevin's)car --> His Kevin's car)
    text = text.replace('?', ' ')            # Removes any ? characters (Example: What?Who said that? --> What Who said that)
    text = text.replace('!', ' ')            # Removes any ! characters (Example: Oh no!What will... --> Oh no What will...)
    text = re.sub(r'\.(?!\d)', ' ', text)    # Removes any . characters unless they are part of a number (Example: Yes.No problem. --> Yes No problem)
    text_clean = re.sub(r'[^\w\s]', '', text) # Removes any remaining punctuation

    return text_clean

# This function appends columns to the dataframe useful for performing a topic analysis
def topic_analysis(dataframe):

    dataframe['tokenized_body_text'] = dataframe['clean_body_text'].apply( lambda x: tokenize( str(x).lower() ) )
    dataframe['body_text_no_stop'] = dataframe['tokenized_body_text'].apply( lambda x: remove_stopwords(x) )
    dataframe['body_text_stemmed'] = dataframe['body_text_no_stop'].apply( lambda x: stemming(x) )
    dataframe['body_text_lemmatized'] = dataframe['body_text_no_stop'].apply( lambda x: lemmatizing(x) )

    return dataframe
```

Figure 8 - Python Code to Preprocess the Post Data

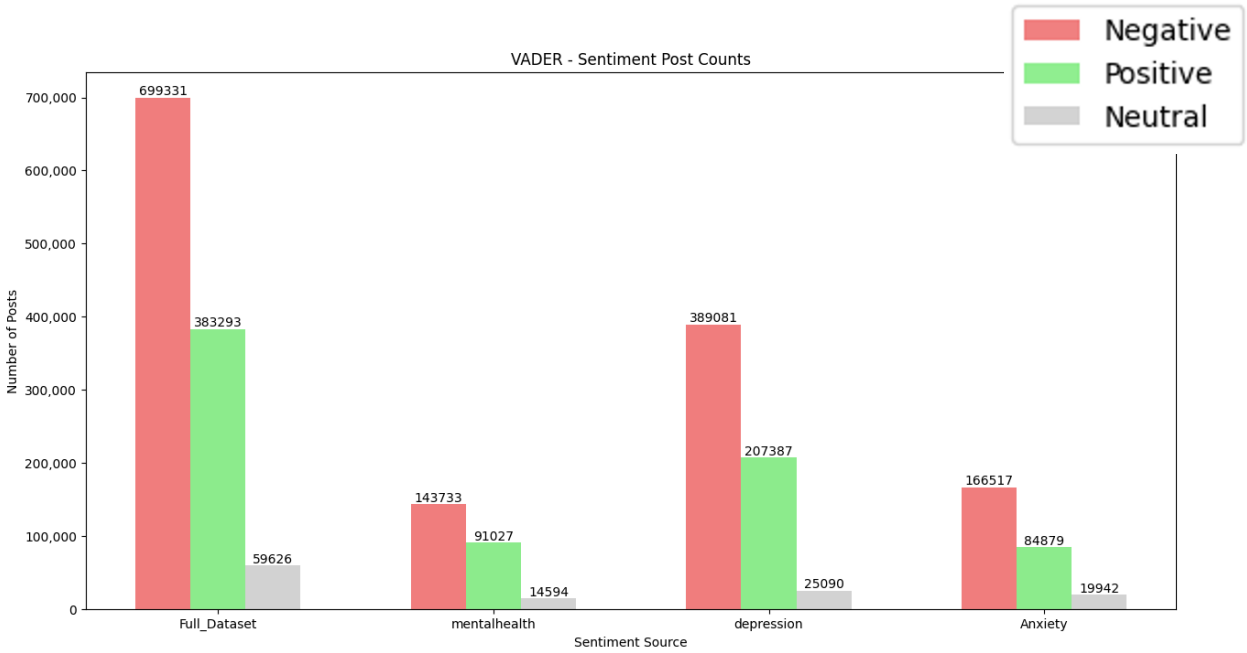
Meanwhile, topic modeling was handled primarily by taking advantage of the stemming and lemmatization facilities offered by Python's Natural Language Toolkit (NLTK), which was used to preprocess the textual data (Figure 7 and Figure 8). Stepwise, this process took the following form:

1. Removing special characters, punctuation, and URLs from the post body text
2. Tokenizing the post body text
3. Removing stopwords from the post body text
4. Stemming the post body text
5. Lemmatizing the post body text

With this done, the most commonly used significant terms could be identified across all of the posts in the dataset, along with the most common flairs used to categorize post content, following the application of similar methods.

Once the results of each project area were determined, graphical representations of sentiment and topic-based totals were produced using the Pyplot API for Matplotlib.

Results



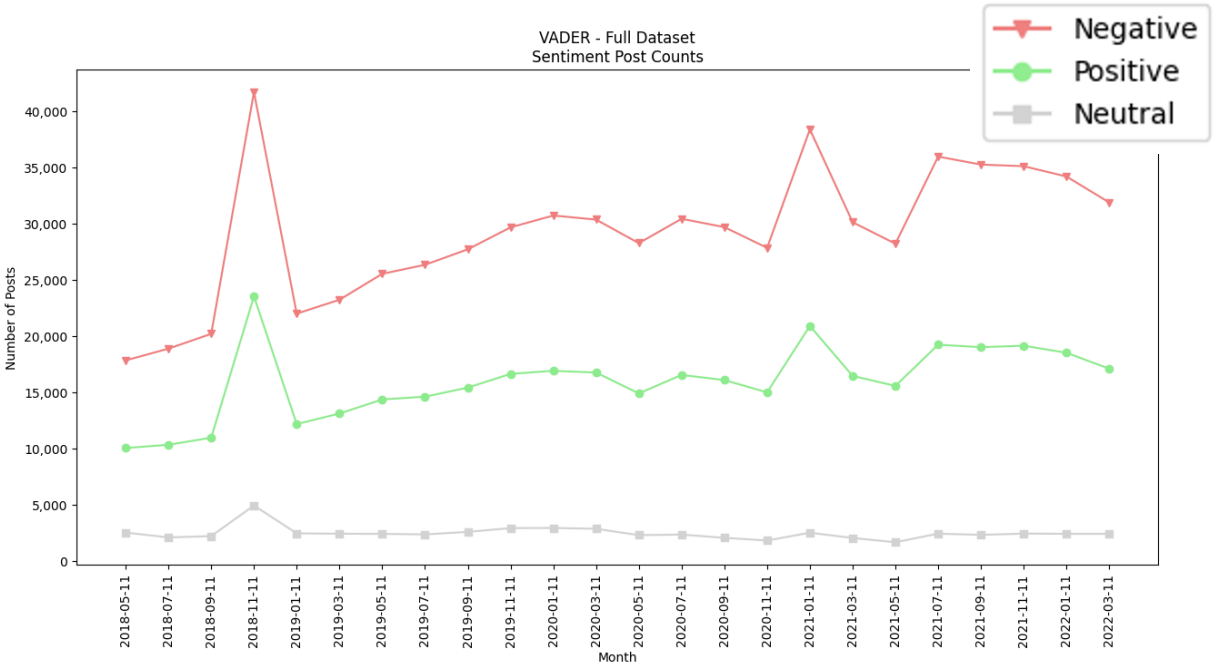


Figure 9 - VADER Sentiment Findings Visualized Using Pyplot

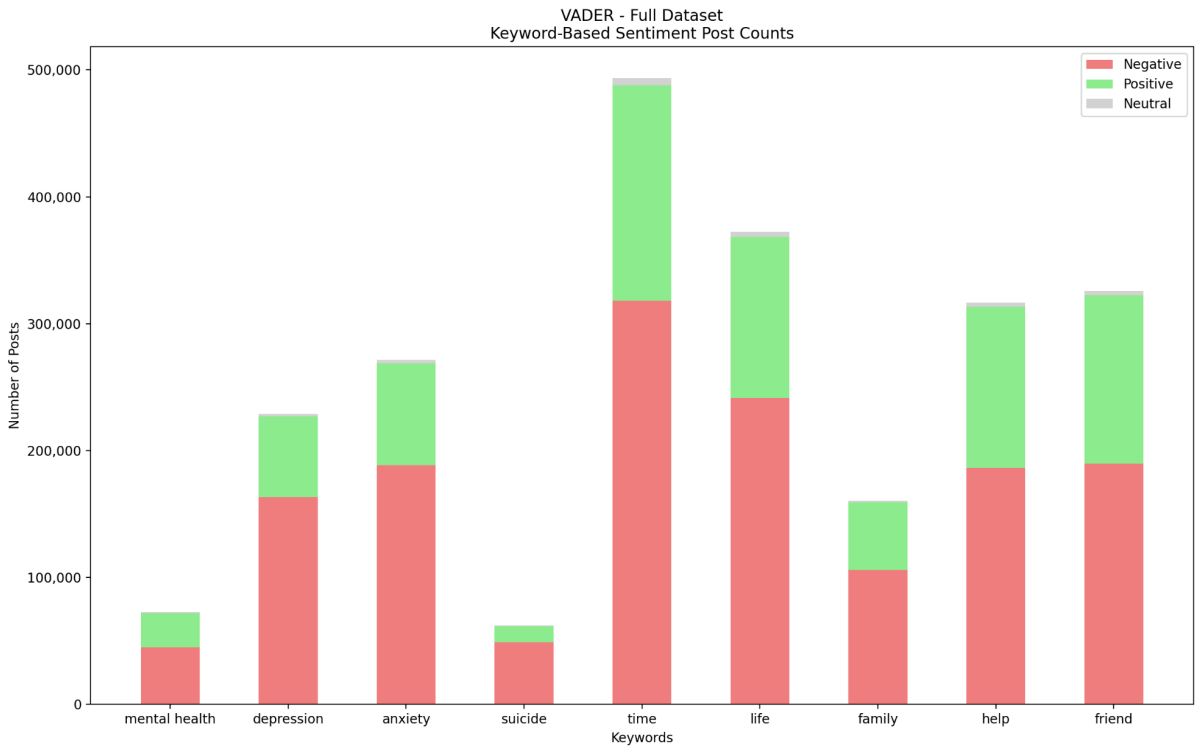


Figure 10 - VADER Keyword-Based Sentiment Findings Visualized Using Pyplot

Once plotted, the preliminary sentiment analysis findings generated using the VADER library indicated that a majority of the dataset's posts could be categorized as negative in tone, both when viewed in total and over time (Figure 9). Furthermore, the totals in particular could also be recontextualized to showcase their relationship with specific keywords related to mental health, based on how many posts they appear in across the full dataset (Figure 10).

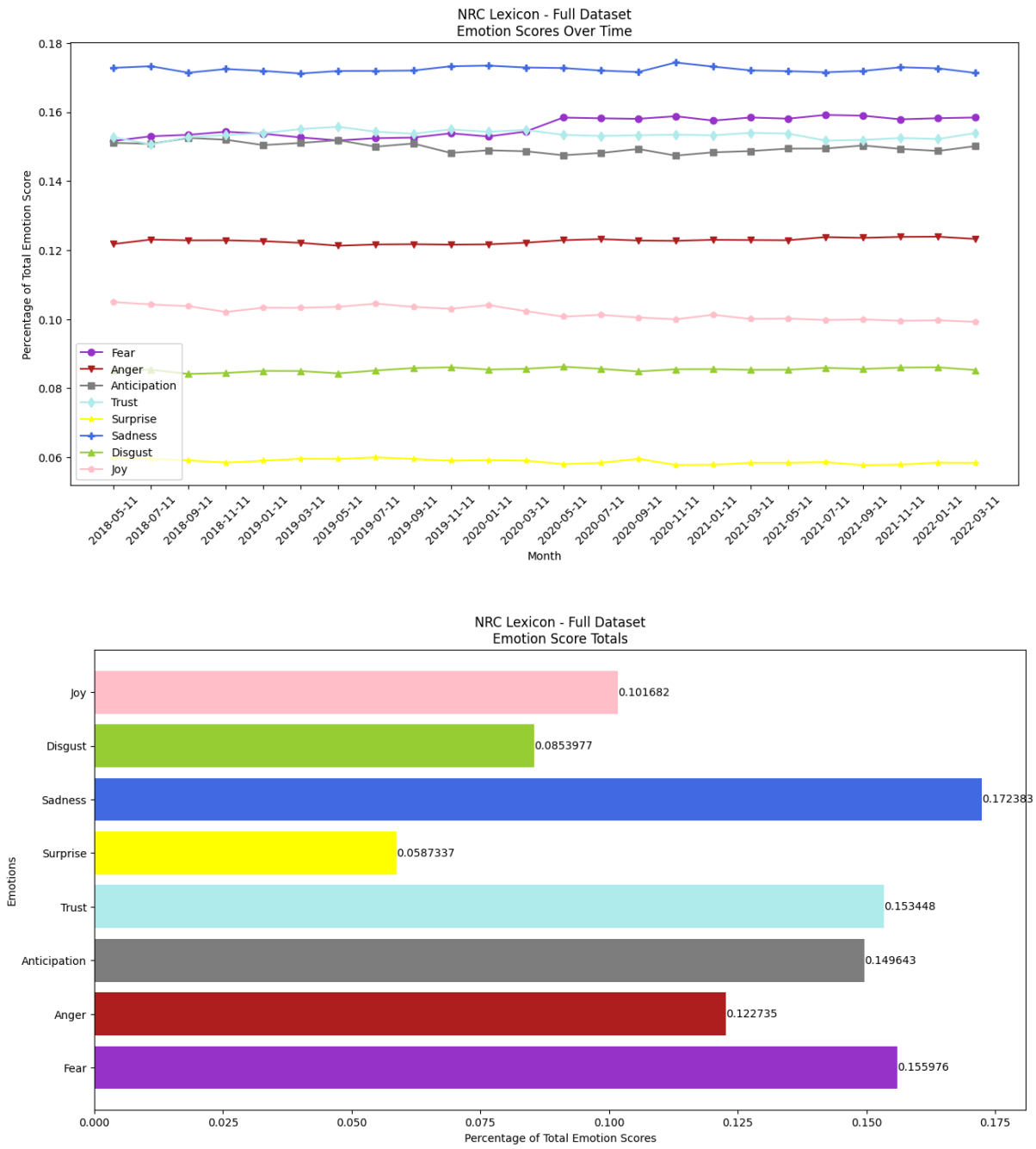


Figure 11 - NRC Lexicon Findings Visualized Using Pyplot

The emotions sadness, trust, anticipation, and fear consistently topped the NRC Emotion Lexicon results, with the latter three swapping positions over time, while a noticeable uptick in fear, occurring right around the beginning of the pandemic on the

overall collection period timescale, is visible at the midway point of the line plot (Figure 11).

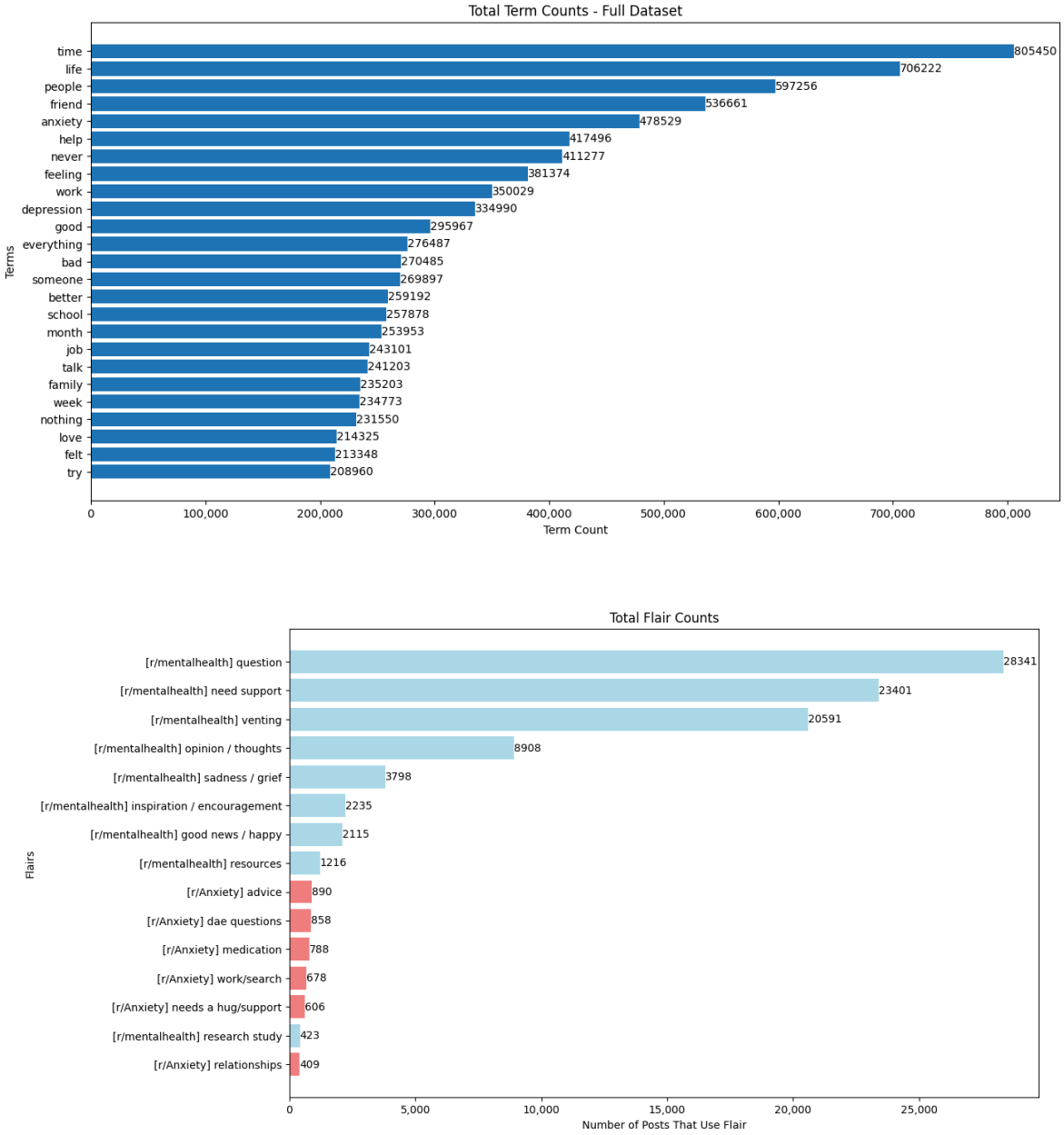


Figure 12 - Topic Model (Term and Flair) Findings Visualized Using Pyplot

As for the topic model, “time”, “life”, “people”, “friend”, “anxiety”, “help”, “never”, “feeling”, “work”, and “depression” rounded out the top ten significant terms identified,

while the flairs “Question”, “Need Support”, and “Venting” accounted for the highest proportion of tags used (Figure 12).

Conclusion

Given the popularity of the website and the size of Reddit’s userbase, analyzing post data from the three target subreddits offers a glimpse into how the subject of mental health is approached by a statistically significant percentage of those inclined to field their discussions of the issue in an online space. With these findings, conclusions can be drawn as to what topics seem to dominate user focus, along with what particular emotional response they are most often eliciting.

References

[1] Dominguez, Hector Rodriguez. “Using Pushshift API for Data Analysis on Reddit.” Medium, MCD-UNISON, 14 Sept. 2021, https://medium.com/mcd-unison/using-pushshift-api-for-data-analysis-on-reddit-b08d339c48b8.

[2] Hutto, C., and E. Gilbert. “VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text”. Proceedings of the International AAAI Conference on Web and Social Media, vol. 8, no. 1, May 2014, pp. 216-25, doi:10.1609/icwsm.v8i1.14550.

[3] Mohammad, Saif M. and Peter D. Turney. “CROWDSOURCING A WORD–EMOTION ASSOCIATION LEXICON.” Computational Intelligence 29 (2013): n. pag.

[4] Mohammad, Saif M. and Peter D. Turney. “Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon.” HLT-NAACL 2010 (2010).