**Application of Time Perspective in Opioid Addiction Study**

Gian Carlo L. Baldonado

CSC 874 Topics in Big Data Analysis

Dr. Rahul Singh

Dec 17, 2022

# Introduction

Opioid addiction continues to be a contemporary health crisis in the United States. In a 2020 study, an estimated 10.1 million people aged 12 or older misused opioids in the previous year [1]. With the societal stigma against opioid misuse, it is challenging to model the epidemic of opioid addiction. In recent years, however, advancements in machine learning and big data can facilitate the understanding and prediction of opioid use through social media posts. In this project, I machine learning, specifically natural language processing (NLP), in extracting time perspective from social media posts. Time perspective (TP) is the manner in which individuals divide the flow of their human experience into different time zones (past, present, future). Social science and psychology studies have shown that a person’s TP has a profound impact on different aspects of their personhood, including their health [2]. Given this context, the use of time perspective in the opioid addiction study can be potentially significant in analyzing the current landscape of opioid use and abuse, which in turn can help inform future public health policies, interventions, and treatment programs. In this paper, I explore how TP can help model the epidemic through natural language processing techniques using Twitter and Reddit social media posts. First, I discuss my problem formulation, followed by an idea of an approach. I will then canvass prior and related work, my methodology and models used (description of work), experimental evaluations, and conclusion and future work.

# Proposed Problem Formulation and Data Collection

Given opioid users' social media posts, I aim to identify and characterize each post of its temporal orientation relative to the user’s opioid addiction state. For each post, my goal is to determine its label: *past*, *present*, *future* (for clarity these temporal orientation classes are italicized hereafter).

The project is based on publicly available data from Twitter and Reddit. For the Twitter data collection, I used tweepy, a Python library for accessing the Twitter API. For the Reddit data, I used PushiftAPI and collected data from two subreddits, r/Opiates and r/OpiatesRecovery. Reddit posts are typically longer than Twitter posts because of Twitter's 280-character limit. To solve this discrepancy, I favored Reddit posts with less than or equal to 280 characters. I amassed a total of 1,010 tweets, 49,452 Reddit posts from r/Opiates, and 13,852 Reddit posts from r/OpiatesReovery. After I independently analyzed and annotated these posts with labels *future, past, and present*, I have a final dataset of 648 annotated social media posts. Evidently, the project does not have any limitations on collecting public data; the constraints were primarily on the data annotation power of one annotator who is a graduate student with novelty in the subject at hand.

# The Idea

My idea has the following steps: (1) *Data collection and data annotation*. In this first step, I plan to analyze Twitter and Reddit social media posts and label them with their temporal orientation, *future*, *past*, and *present*, relevant to their opioid addiction state. (2) *Data preprocessing and feature engineering*. Using multiple text pre-processing and analysis libraries, I plan to clean the raw posts with regex and Python’s built-in string library. I will then generate numerical features based on the raw and clean posts using NLTK, TextBlob, StanfordCoreNLP, Scikit-learn, and gensim. (3) *Building models.* I will use Scikit-learn’s off-the-shelf algorithms Multinomial Naive Bayes, Logistic Regression, Random Forest, and Support Vector Machine to attempt to model the problem. (4) *Evaluation*. I will try a total of four experiments per algorithm and compare model performances using evaluation metrics precision, recall, f1, and accuracy scores. With the optimal model, I will perform an error and strength analysis to determine the strengths and weaknesses of the approach.

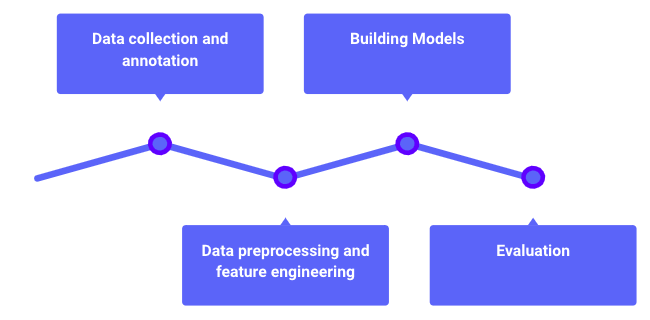
# Prior and Related Work

Among prior works, I read the papers *Identifying and Characterizing Opioid Addiction States Using Social Media Posts* (D. Jha, et. al.) and *Resolution of grammatical tense into actual time, and its application in Time Perspective study in the tweet space* (Sabyasachi Kamila et. al.). In the first paper, the authors extracted Reddit posts to use for identifying the progression of individuals through stages of opioid addiction. This paper advances the state-of-the-art, including the author’s use of word embeddings with bi-directional long short-term memory recurrent neural networks (Bi-LSTM) [2]. In the second paper, Sabyasachi Kamila et. al. collected Twitter data and built a temporal orientation classifier using Bi-LSTM in conjunction with an attention mechanism that can resolve syntactic tense to semantic time [3]. These papers have inspired my work to leverage time perspective in the context of the opioid crisis.

My work attempts to be foundational in the application of time perspective in the study of opioid addiction. Specifically, my work presents findings in the use of traditional machine learning algorithms, and how they perform in identifying temporal orientations given a relatively small dataset.

# Description of Work

My work follows the same steps as my idea: (A) *Data collection and annotation*, (B) *Data preprocessing and feature engineering*, (C) *Building models*, and (D) *Evaluation.*



**Figure 1:** The timeline of building a temporal orientation classifier.

## Data Collection and Annotation

As aforementioned, the dataset was collected from Twitter and Reddit using tweepy and PushiftAPI. A total of 64,314 social media posts were collected: 1,010 tweets, 49,452 Reddit posts from r/Opiates, and 13,852 Reddit posts from r/OpiatesReovery. After analyzing thousands of these data points, the final dataset consists of 648 annotated social media posts.

For the Twitter data collection, I used opioid-related keywords from another study[4].

| **Keyword** | **Common Misspellings** |
| --- | --- |
| Tramadol | trammadol tramadal tramdol tramadols tramado tramedol tramadoll tramadole tramidol tamadol tranadol tramodol tremadol |
| Heroin | herione herroine heroins heroine heroin heorin herion |
| Methadone | methadones methadose methodone mehtadone metadone methadon methdone |
| Oxycontin | oxicontin oxcotin oycotin oxycotins oycontin oxycontins oxycoton oxicotin ocycotin oxycodin oxycottin oxycotine ocycontin |
| Codeine | codiene coedine codine codene codein |
| Dilaudid | delaudid dialudid dilaudad diluadid diaudid dilaudin dilauded dilauid dillaudid |

**Table 1:** Sample of opioid-related keywords and their common misspellings used in the Twitter data collection.

Of the final dataset, there are 140 (21.6%) future-oriented posts, 240 (37.0%) past-oriented posts, and 268 (41.4%) present-oriented posts. In Table 3, I present a further breakdown of the dataset across training and test set.

| **TP Orientation** | **Sample Post** |
| --- | --- |
| Future | I’m like 20-ish days clean and the Brain zaps finally stopped.. my goal is to not use any hard drugs all of 2021 let’s make it happen 🙏🏽 |
| Past | I was on 170mg and had been for two years. The physical withdrawals lasted a little over a month, the mental stuff went away after about 3. |
| Present | I am on day 8 as well and the mood swings are insane. I thought I was going crazy. |

**Table 2:** Sample of social media posts and their temporal orientation label.

In determining the temporal orientation of the posts, I created and followed these guidelines to determine the TP classification of the post:

1. Label a social media post as *future* if the user is motivated by the anticipation of tomorrow or the underlying temporal connotation of the tweet refers to the *future* time relevant to the user’s opioid addiction state.
2. Label a social media post as *past* if the user is motivated by memories of the past or the underlying temporal connotation of the tweet refers to the *past* time relevant to the user’s opioid addiction state.
3. Label a social media post as *present* if the user is motivated by their current situation or the underlying temporal connotation of the tweet refers to the *past* time relevant to the user’s opioid addiction state.

## Data Preprocessing and Feature Engineering

* 1. **Data Preprocessing**

On each social media post, I performed the following pre-processing tasks: (1) *De-constructing contractions*. The informality of social media posts includes the use of contractions, the combination of two or more words in a shortened form (e.g. *I’ll* is the contraction of *I will*). I manually created a few of the most common contractions as rules with regex. (2) *Performed basic substitutions and cleaning*. This step includes converting *\n* and *\t* to white spaces, lowercasing the text, removing punctuations, stripping the text from extra white spaces, and removing hyperlinks. (3) *Tokenization.* I then separated the words of each post into a list. This step acts as a helper function to step four. (4) *Removing stopwords.* I modified the stopwords of NLTK, removing verbs and phrases that I determined might be useful for prediction (e.g. I removed the word *won’t* in the stopwords because it may add context in determining the TP of the post). After preprocessing, I then split the dataset into training and testing sets.

|  | **Training set**  **(n=434)** | **Testing set**  **(n = 214)** |
| --- | --- | --- |
| **TP Orientation** | **Count (%)** | **Count (%)** |
| Future | 90 (20.74) | 50 (23.36) |
| Past | 163 (37.56) | 77 (35.98) |
| Present | 181 (41.71) | 87 (40.65) |

**Table 3:** Breakdown of the composition of the training and test sets.

* 1. **Feature Engineering**

For feature engineering, I employed a mixture of text analysis libraries to obtain numerical features and manually crafted features as informed by my prior work. A feature-engineered dataset that includes all features has a total of 367 features.

### Polarity and subjectivity scores

For polarity and subjectivity, I used the uncleaned text as inputs because punctuations, letter cases, and emojis can help in identifying the feature. I used the NLTK sentiment analyzer called Vader. Vader uses a compound score that calculates the sum of all the lexicon ratings normalized between -1 (extremely negative) to 1 (extremely positive) [5]. Vader’s polarity output presents four scores: a negative score (compound score >= 0.05, a neutral score (compound score >-0.05 and compound score < 0.05), a positive score (compound score <= 0.05), and the compound score. For subjectivity, I used TextBlob’s subjectivity method which looks into the intensity of words in the text. Its score ranges from 0 (least subjective) to 1 (most subjective).

### Time expressions count

Time expressions are words or phrases that communicate a period of time, the duration of something that happens, such as *now*, *today*,and *February 28, 2020.* Per my prior work, I determined counting temporal expressions may help in the classification task. I used StanfordCoreNLP’s SUTime, a library for processing temporal expressions.

### Temporal modifiers count

Per my prior work and data collection, I noticed that in some of the posts that I have read, the temporal modifiers that exist in those posts have helped me identify their temporal orientation. With this in mind, I created a list of temporal adjectives and adverbs and classified them by their temporal type.

| **Temporal Type** | **Adverbs and Adjectives** |
| --- | --- |
| Futuristic | finally, later, next, soon, yet, forward |
| Past | then, before, formerly, last, late, previously, past, former |
| Past-present | yesterday, already, early, earlier, since, earlier |
| Present | now, today, just, lately, recently |
| Present-futuristic | tomorrow, tonight, later |
| Undetermined | daily, fortnightly, hourly, monthly, nightly |

**Table 4**: Some of the temporal adverbs and adjectives used in feature engineering.

### Part-of-Speech tag count

In addition to the temporal modifiers count, I incorporated a general part-of-speech tag feature that counts the instances of nouns, verbs, interjections, adjectives, and adverbs (not limited to temporal modifiers). For this task, I used NLTK part-of-speech tagging.

### Word2Vec and TF-IDF

Finally, I used two different word representation techniques to obtain word associations. For Word2Vec, I accounted for bi-grams and tri-grams and I created a word2vec model with 300 dimensions. For TF-IDF, I used Scikit-learn’s TFIDFVectorizer with max\_features of 10,000, and an n-gram range of (1,2). I decreased the dimensionality of the overall TFIDF sparse matrix using Scikit-learn’s chi-square feature selection with a p-value limit of 0.80. From 10,000 features, I reduced the TFIDF matrix to 46 most relevant features.

## Building Models

For the model creation, I compared and used four traditional statistical algorithms: Multinomial Naive Bayes, Logistic Regression, Random Forest, and Support Vector Machine. For each algorithm, I ran 4 different models:

* Model 0 uses the following features hereby known as “simple features”: polarity and subjectivity scores, time expression count, temporal modifier count, and part of speech count.
* Model 1 uses all the features: simple features, the Word2Vec matrix, and the TF-IDF matrix.
* Model 2 uses simple features and Word2Vec.
* Model 3 uses simple features and the TF-IDF matrix.

I performed grid-search cross-validation to obtain the optimal hyperparameters for each model. The following table displays the hyperparameters I used for each algorithm.

| **Algorithm** | **Dictionary of Hyperparameters** |
| --- | --- |
| Multinomial Naive Bayes (NB) | {  'alpha' : np.linspace(0.5, 1.5, 6),  'fit\_prior' : [True, False]  } |
| Logistic Regression (LR) | {  'C' : [0.1, 1, 10, 100],  'solver' : ['lbfgs', 'liblinear'],  'penalty': ['l1', 'l2'],  'max\_iter': [200,400,600]  } |
| Random Forest (RF) | {  'n\_estimators' : [100,200,500,1000],  'min\_samples\_leaf':[1, 2, 5, 10,20],  'max\_features' : [int(0.5\*np.sqrt(len\_features)), int(np.sqrt(len\_features)), int(2\*np.sqrt(len\_features))]  } |
| Support Vector Machine (SVM) | {  'kernel': ['rbf', 'linear'],  'gamma': [1e-3, 1e-4],  'C': [1, 10, 100, 1000]  } |

**Table 5**: Algorithm and hyperparameters used in GridSearchCV.

## Experimental Evaluations

* 1. **Results of the GridSearch Cross Validation: Optimal Hyperparameters**

| **Model 0: (Simple Features) Polarity + Subjectivity Score, Count of Temporal Expressions, Count of Temporal Modifiers, POS Tags** | |
| --- | --- |
| NB | {'alpha': 0.7, 'fit\_prior': False} |
| LR | {'C': 1,  'max\_iter': 200,  'penalty': 'l1',  'solver': 'liblinear'} |
| RF | {'max\_features': 2,  'min\_samples\_leaf': 2,  'n\_estimators': 500} |
| SVM | {'C': 1000, 'gamma': 0.001, 'kernel': 'linear'} |
| **Model 1: (All features) simple\_features + tfidf + w2v** | |
| NB | {'alpha': 1.1, 'fit\_prior': False} |
| LR | {'C': 1,  'max\_iter': 200,  'penalty': 'l2',  'solver': 'liblinear'} |
| RF | {'max\_features': 8,  'min\_samples\_leaf': 1,  'n\_estimators': 100} |
| SVM | {'C': 1, 'gamma': 0.001, 'kernel': 'linear'} |
| **Model 2: Simple features + w2v** | |
| NB | {'alpha': 1.5, 'fit\_prior': True} |
| LR | {'C': 1,  'max\_iter': 400,  'penalty': 'l2',  'solver': 'lbfgs'} |
| RF | {'max\_features': 8,  'min\_samples\_leaf': 2,  'n\_estimators': 200} |
| SVM | {'C': 1, 'gamma': 0.001, 'kernel': 'linear'} |
| **Model 3: Simple features + tfidf** | |
| NB | {'alpha': 0.5, 'fit\_prior': False} |
| LR | {'C': 100,  'max\_iter': 200,  'penalty': 'l2',  'solver': 'liblinear'} |
| RF | {'max\_features': 4,  'min\_samples\_leaf': 1,  'n\_estimators': 500} |
| SVM | {'C': 10, 'gamma': 0.001, 'kernel': 'linear'} |

**Table 6**: Optimal hyperparameters determined by GridSearchCV across algorithms and across models.

* 1. **Model and Algorithm Performance Comparison**

| **Algorithm** | **Model** | **Precision** | **Recall** | **F1** | **Accuracy** |
| --- | --- | --- | --- | --- | --- |
| Random Forest | 0 | 0.54 | 0.46 | 0.44 | 0.51 |
| Random Forest | 1 | 0.59 | 0.56 | 0.55 | 0.56 |
| Random Forest | 2 | 0.53 | 0.51 | 0.49 | 0.51 |
| Random Forest | 3 | 0.53 | 0.52 | 0.50 | 0.52 |
| Logistic Regression | 0 | 0.49 | 0.46 | 0.43 | 0.51 |
| Logistic Regression | 1 | 0.54 | 0.54 | 0.54 | 0.54 |
| Logistic Regression | 2 | 0.51 | 0.52 | 0.51 | 0.52 |
| Logistic Regression | 3 | 0.54 | 0.53 | 0.53 | 0.53 |
| Naive Bayes | 0 | 0.45 | 0.45 | 0.44 | 0.50 |
| Naive Bayes | **1** | **0.59** | **0.59** | **0.59** | **0.59** |
| Naive Bayes | 2 | 0.53 | 0.54 | 0.52 | 0.54 |
| Naive Bayes | 3 | 0.58 | 0.57 | 0.57 | 0.57 |
| SVM | 0 | 0.45 | 0.47 | 0.43 | 0.53 |
| SVM | 1 | 0.53 | 0.53 | 0.53 | 0.53 |
| SVM | 2 | 0.53 | 0.54 | 0.53 | 0.54 |
| SVM | 3 | 0.54 | 0.53 | 0.53 | 0.53 |

**Table 7**: Comparison of the different algorithms across different models. The optimal model is Naive Bayes in conjunction with Model 2.

Based on Table 7, it is clear that Model 1 Naive Bayes outperforms the other models and algorithms, with Model 3 Naive Bayes as the second-best performing, and Model 1 Random Forest as the third-best performing. Unfortunately, the results, including the scores of the optimal model, are poor, and I would not recommend the use of any of these models for eliciting a data point for decision-making. However, the comparisons of the models show a promising sign that further work with the models, including the use of more data points, the use of different algorithms such as Bi-LSTM (which is designed for sequential data like text), and a thorough review of the data collection and annotation process, may improve the performance of the models. With this in mind, I performed an error analysis that may inform future work on the models.

* 1. **Error Analysis**

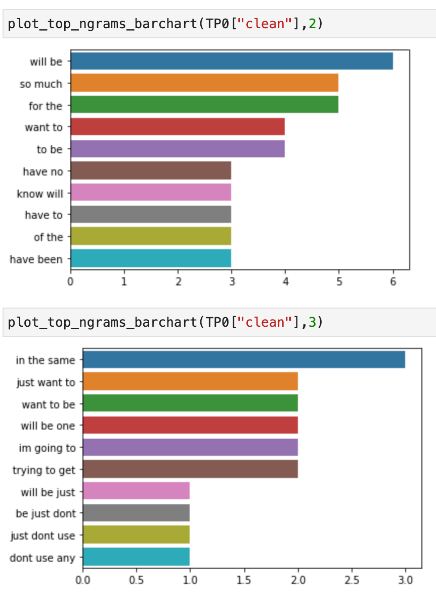
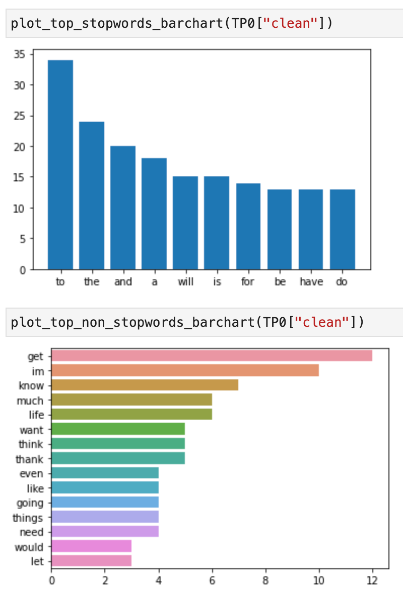
I further analyzed the optimal model’s (Naive Bayes Model 1) strengths and weaknesses using confusion matrices. Because there are three classes, there are 9 different classification states that a data point can be identified. I assign the following nomenclature to these states.

* TP0: correctly identified *future* posts
* TP1: correctly identified *past* posts
* TP2: correctly identified *present* posts
* E01: *future* posts misclassified as *past* posts
* E02: *future* posts misclassified as *present* posts
* E10: *past* posts misclassified as *future* posts
* E12: *past* posts misclassified as *present* posts
* E20: *present* posts misclassified as *future* posts
* E21: *present* posts misclassified as *past* posts

| **State** | **Count** |
| --- | --- |
| TP0 | 25 |
| TP1 | 50 |
| TP2 | 51 |
| E01 | 5 |
| E02 | 20 |
| E10 | 9 |
| E12 | 18 |
| E20 | 13 |
| E21 | 23 |

**Table 8**: Comparison of the different algorithms across different models. The optimal model is Naive Bayes in conjunction with Model 2.

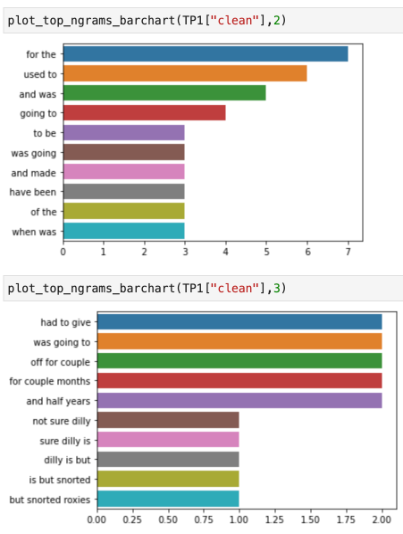
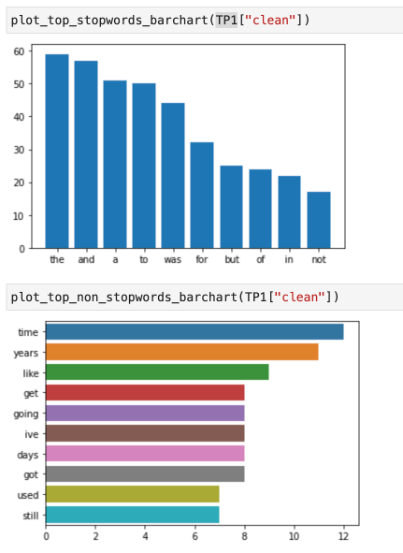
#### **TP0**



**Figure 2**: (Top left to bottom right) Data analysis of word frequencies and n-grams for TP0.

Correctly labeled *future* posts are characterized by n-grams “will be”, “so much”, “want to”, “know will”, etc.

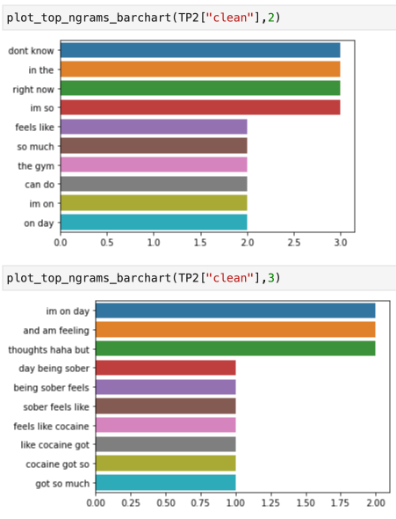
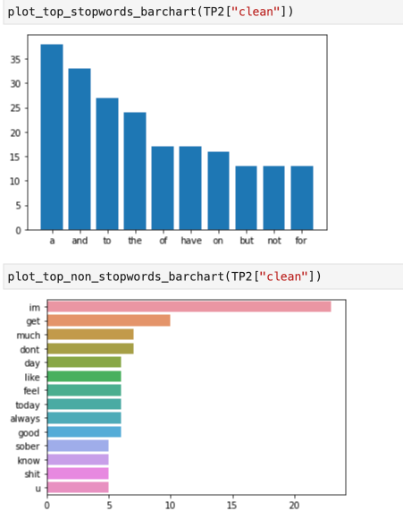
1. **TP1**

****

**Figure 3**: (Top left to bottom right) Data analysis of word frequencies and n-grams for TP1.

Correctly labeled *past* posts are characterized by n-grams “used to”, “and was”, “was going”, “have been”, etc.

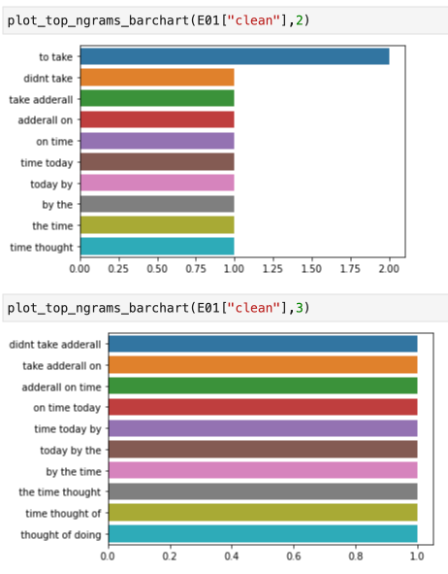
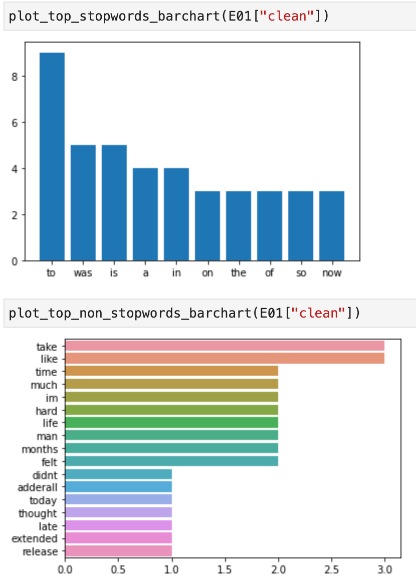
1. **TP2**

****

**Figure 4**: (Top left to bottom right) Data analysis of word frequencies and n-grams for TP2.

Correctly labeled *present* posts are characterized by “right now”, “dont know”, feels like”, “on day”, etc.

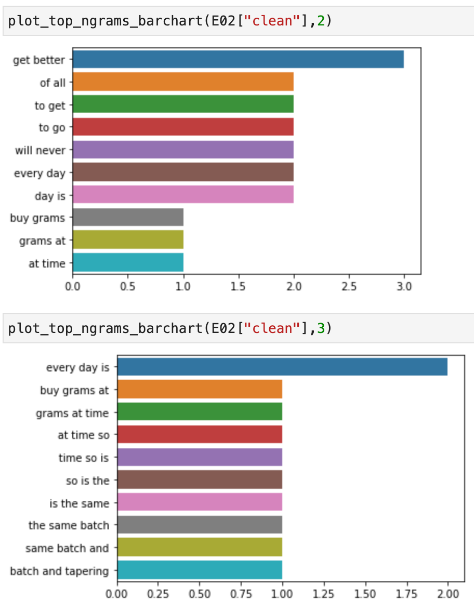
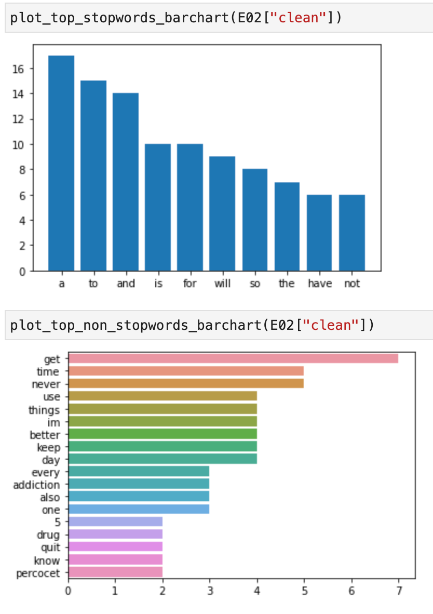
1. **E01**

****

**Figure 5**: (Top left to bottom right) Data analysis of word frequencies and n-grams for E01.

Hypothesis: These *future* posts have keywords and n-grams and bigrams that are common in *past* posts.

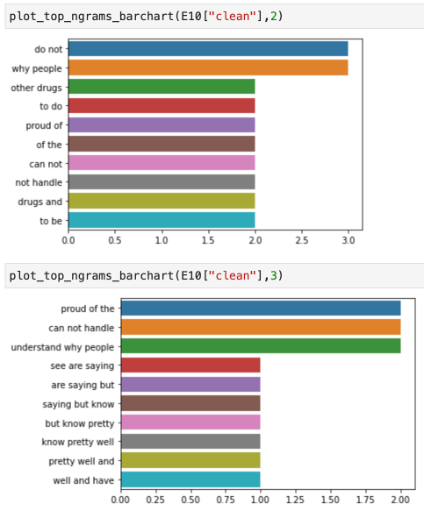
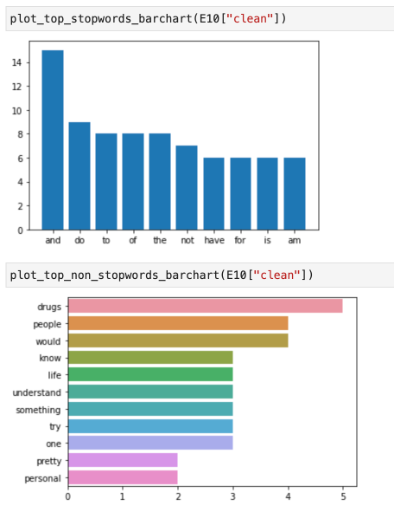
1. **E02**

****

**Figure 6**: (Top left to bottom right) Data analysis of word frequencies and n-grams for E02.

Hypothesis: A lot more of the *future* posts are misclassified as *present* posts. There is a better separation of *future* and *past* posts. My hypothesis is that linguistically, future-oriented posts tend to have present tense verbs that are compounded with auxiliary verbs like "will" and "shall" to make the future. There is no morphological difference, verb-wise, in separating future and *present* posts, unlike present and past tense which has morphological differences in verb tense.

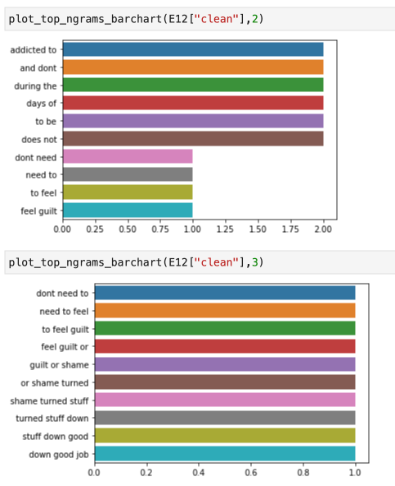
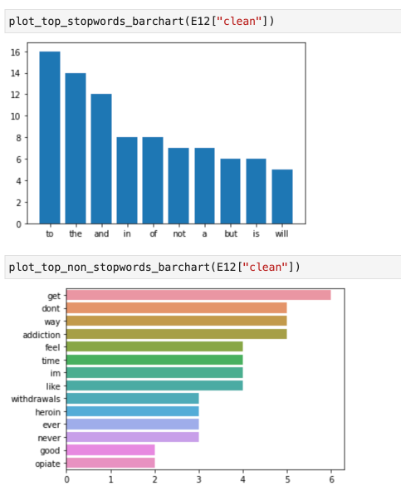
1. **E10**

****

**Figure 7**: (Top left to bottom right) Data analysis of word frequencies and n-grams for E10.

Hypothesis: These *past* sentences are mostly multi-sentence with a good percentage of *future*-orientedness (n-grams common in future-orientedness) and therefore could have affected the model's interpretation of the sentence.

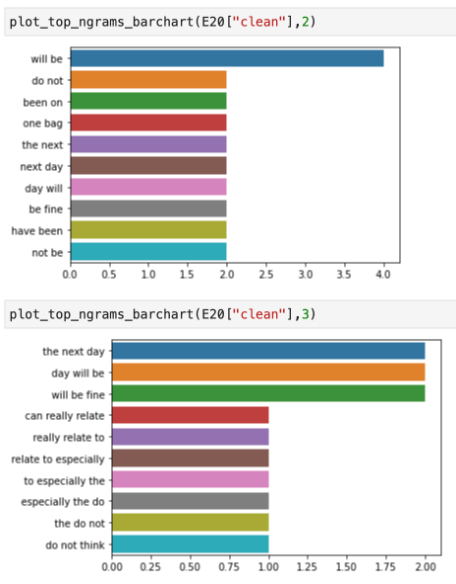
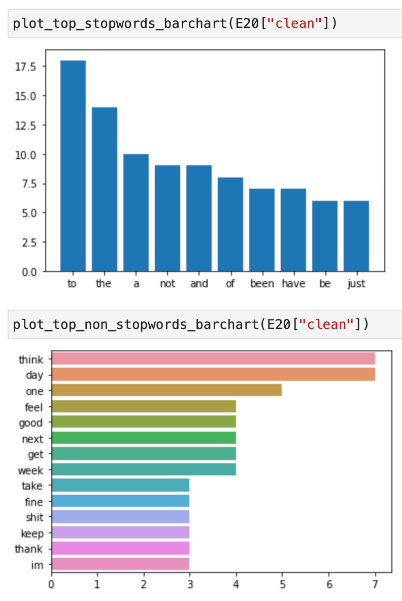
1. **E12**

****

**Figure 8**: (Top left to bottom right) Data analysis of word frequencies and n-grams for E12.

Hypothesis: We are analyzing *past*-oriented posts misclassified as *present*-oriented. Similar to e10, the documents used are multi-sentences and therefore have qualities of different temporal orientations. While *past* and *present* have morphological differences, it is important to note that verb tense is simply one criterion that was looked at in the methodology.

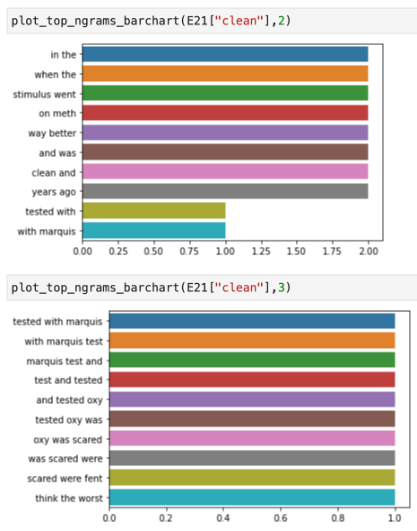
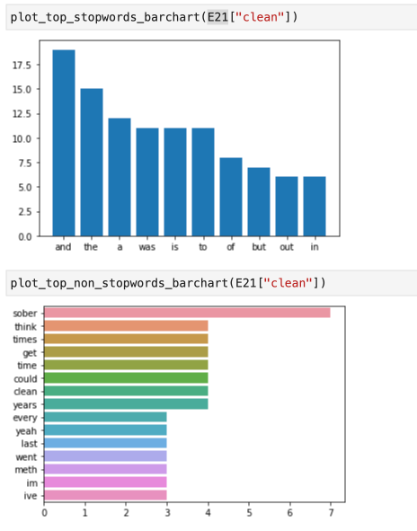
1. **E20**

****

**Figure 9**: (Top left to bottom right) Data analysis of word frequencies and n-grams for E20.

Hypothesis: These *present*-oriented sentences are misclassified as *future*-oriented because they carry strong *future*-oriented n-grams.

1. **E21**

****

**Figure 10**: (Top left to bottom right) Data analysis of word frequencies and n-grams for E21.

Hypothesis: A lot more *present*-oriented posts are misclassified as *past* compared to future misclassification. Similar to E20, E21 sentences are complex and carry *past*-oriented features such as the presence of *past* tense verbs and n-grams.

# Conclusion

To conclude, while the performances of the models were poor and are not recommendable for use in any system, the comparison of the different algorithms using different models (features) shows a sign of a gradual increase in performance when all features are used with Naive Bayes. Because of the class independence assumption of Naive Bayes, it performs well with relatively small datasets as it can quickly learn high-dimensional features. I then performed an error analysis on the optimal model (Naive Bayes Model 1) to see which data points the model struggled with the least and most. It was expected that the model will perform well in identifying *present* posts compared to *future* posts because there were more *present* than *future* posts in the training data. Based on the error analysis performed, most of the errors may stem from two aspects: the multi-sentence structure of the posts and the presence of verb tenses that may be associated with *past*, *present*, and *future*. Multi-sentences can complicate the identification of the post’s TP as there are more words to account for, including a mixture of different tenses that can further complicate the prediction. For future work, I aim to use more data points as I believe this may be one of the limitations of the current approach. With more data points, I can leverage deep learning techniques like Bi-LSTM, which has context management for long-range dependencies of sequential data. Furthermore, further exploration of the error analysis, exploratory data analysis, and a review of the data annotation and collection may help the performance of the current approach.

# References

1. Substance Abuse and Mental Health Services Administration. (2020). Key substance use and mental health indicators in the United States: Results from the 2019 National Survey on Drug Use and Health. (HHS Publication No. PEP20-07-01-001(NSDUH Series H-55). https://www.samhsa.gov/data/
2. D. Jha, S. R. La Marca and R. Singh, "Identifying and Characterizing Opioid Addiction States Using Social Media Posts," 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2021, pp. 913-918, doi: 10.1109/BIBM52615.2021.9669628.
3. Kamila, S., Hasanuzzaman, M., Ekbal, A., & Bhattacharyya, P. (2019). Resolution of grammatical tense into actual time, and its application in Time Perspective study in the tweet space. *PLOS ONE*, *14*(2), e0211872. https://doi.org/10.1371/journal.pone.0211872
4. Sarker, A., Gonzalez-Hernandez, G., & Perrone, J. (2019). Towards automating location-specific opioid toxic surveillance from Twitter via data science methods. *Studies in health technology and informatics*, *264*, 333. <https://doi.org/10.3233/SHTI190238>
5. Geeks For Geeks. (2021, October 7). Python | Sentiment Analysis using VADER. GeeksforGeeks. Retrieved December 17, 2022, from <https://www.geeksforgeeks.org/python-sentiment-analysis-using-vader/>