CS 4375 Final Project Report: Textual Classification of Political Tweets on the Political Spectrum

Fawaz Ahmed, fxa180017, fawaz.ahmed@utdallas.edu, CS Department;
Ivan Payne, iap170000, iap170000@utdallas.edu, CS Department

May 21, 2022

Abstract

Social Media, especially Twitter, is a very useful tool for discussing with other people about politics. Its popularity results in vast amounts of tweets or a post on Twitter, including the user's political stance on certain events. In this paper, we develop two models. The first is a model that can identify whether a given tweet is political with an accuracy of around 90%. This model was trained using a dataset provided by Micol Marchetti-Bowick and Nathanael Chambers and contains around 4 thousand instances of data. This model aimed to identify whether a tweet was political or not once we collected it via the Twitter API. The second model can classify a political text as either "left-leaning" or "right-leaning" with an accuracy of around 89%. This model was trained on a dataset of around 3 thousand tweets with a split of 1900 being left-leaning and 1200 being right-leaning. These tweets were collected via the Twitter API from politically active users. The results demonstrate that it is possible to classify tweets on a political spectrum.

1 Introduction

Social media shapes how information, specifically politically-related information, is shared. It allows people to share their opinions on current events on a relatively anonymous platform. This allows users to be more open about what they feel as a semi-anonymous environment lets people express themselves[1]. If we can analyze and collect this data, we can predict the response to a specific event. This motivated us to create a model that can predict a tweet's political stance as it can be used to identify which type of people care about this event the most and how they react to it.

Twitter, a social media platform, provides users with a semi-anonymous platform to share their opinions. This platform has been used in the past as a place to predict political events like an election outcome in Indonesia[2]. Twitter has also been used in the 2012 election to predict the public's sentiment towards the political candidates[3]. Research shows that Twitter acts as a center for political information and opinions. With a community of over 330 million active monthly users, there is a lot of information that can be used for political analysis[4]

With an ever-changing political climate, it is vital for those representing the people to know how the people feel about specific issues. However, with how vast the internet is and how many users there are on Twitter, it is easy to get lost in an echo chamber of similar views. Currently, there are limited tools for quickly and accurately measuring the public's views. With the lack of fast information, it becomes hard for Politicians to represent and support the people that voted them into office.

One way to correct this is to connect to this significant information hub. We might know there is outrage or expressed opinions about a political topic, but no person can go through 330 millions users and learn their stance. The goal is to use the Twitter API to grab live tweets from the platform and classify the stances of the public on issues. Separate them into their categories and see how many tweets are supporting the Left or Right stance on current issues.

In this paper, we describe the textual classification and pre-processing methods used on our dataset. We use three Machine Learning algorithms (random forest, naive bayes, and SVM) to train both models. The first model classifies tweets as either political/non-political or 1/0, respectively. The second model classifies a political tweet as either "left-leaning"/"right-leaning" or 0/1, respectively. Each Tweet had to go through several pre-processing steps before it was ready to be used in training/testing. For example, removing emojis, correcting spelling, and lemmatizing the tweets. The process and results for each method will be explained in further detail in the

paper.

Figure 1 illustrates our the framework for our research. We first use the Twitter API to collect tweets for our dataset. We then pre-process the text and finally train our models using the three different algorithms we chose.

This paper is organized as follows, Section 2 contains related research done and its summary. It also talks about the political dataset used to generate model 1. Finally, how this research is related to what we are doing. Section 3 showcases our labeled dataset and our decision for labeling data a certain way or what type of keywords we noticed were being used the most. It also discusses the different topics and users we collected data from and what we noticed about the data. Section 4 will explain the results of our work, including the dataset, training details, and our evaluation of the resulting models. Finally, Section 5 will summarize our work and the results.

2 Related Work

In this section we will describe the related works for the textual classification for political tweets. It will be split into three parts, the first is the pre-processing work, the second is the model to identify if a tweet is political or not, the third is a model for identifying the political leaning for a tweet.

2.1 Pre-processing

Lets say we are given the following tweet

```
Our current healthcare system is
cruel and dysfunctional. Healthcare
is a human right, not a privilege.
We need Medicare for All.
```

This tweet text cannot be processed by a ML algorithm directly and needs to go through several stages of pre-processing. This is an interesting problem and is something that is dealt with by the NLP research community. In this sub-section, we will introduce several research papers that attempt to properly pre-process tweets with as little loss of data as possible. For example, one research paper aims to use a classification model called BERT to pre-process tweets. This type of model is very advanced and can use emojis, emoticons, and hashtags to guess the tone of a tweet[5]. This is useful as when pre-processing text, the emotion, and tone of the user are often lost. Other notable research includes, for example, using Ekman's six basic emotions[6] to classify tweets for pre-processing[7]. In [8], the authors went in-depth into different types of pre-processing for tweets that would be used for sentimental analysis. For example, it talks about how authors implemented a technique to pre-process tweets using slang language/abbreviations analysis, lemmatization, and stop word removal[9]. While others have encoded slang, we found it best to avoid encoding interpretations. The reasoning is that slang is interpreted differently in each social circle. If a spell checker could find an alternative, then it would be replaced. If not, it would be left in to be added to a data map. The determination of whether slang was left or right-leaning would be determined by the occurrence of the word rather than the interpretation of the creators, who could view it differently.

2.2 Political Text Prediction

This sub-section will showcase other research papers that attempted to predict whether or not a tweet or text is politically related. One such research is [10] that analyzed the sentiment toward Barack Obama during his presidency. They used Twitter to collect political tweets related to Obama's presidency. They collected tweets using Twitter's API and used specific keywords related to politics. The tweets collected were then labeled as either political or apolitical. This model had a precision of around 90%. The ML strategy they used is called distant supervision, and it aims to use noise as a way to classify that data. In [11] the authors utilize a previously annotated dataset to train a model that could categorize text into several labels like atheism, climate change, abortion, etc. It used stance classification using deep-learning models on the data collected from Twitter. They trained two neural networks, the first being a long-short term memory (LSTM) network and the second being a convolutional neural network (CNN).

2.3 Political Leaning classification

In this sub-section, we will discuss research that went into classifying a user's text into different political parties/stances. In [12] the authors attempt to classify articles as either liberal or conservative. They applied semi-supervised learning methods that classify an article or user as liberal or conservative. They took sites and blogs with clearly labeled sides to create a baseline for classification. From these sites, they looked at the user, sources, and story itself to build further classification. One issue they find is that people and stories often aren't just one-sided and do contain

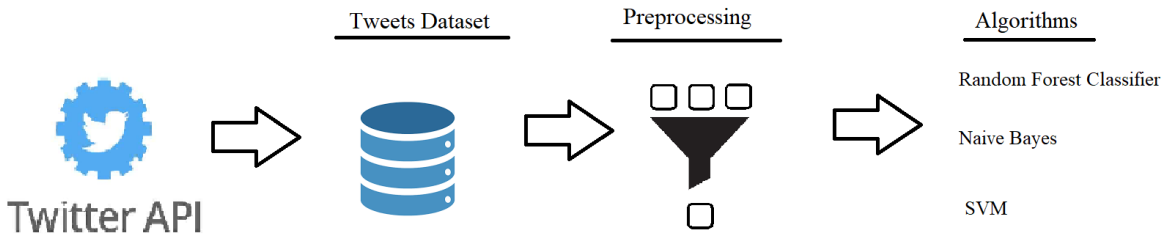

Figure 1: Framework of our research

grey areas. In our case, instead of websites, we led with very vocal leaders of the left and the right as baselines. From here followers, people who liked were followed to expand the baseline of what is left or right-leaning. If something could not be identified to one side, they were immediately determined non-political. We follow the same issue of being able to improve the algorithm's ability to separate clearly left and right tweets from those that do not fit neatly into either category to create a spectrum.

3 Dataset

3.1 Model 1

3.1.1 Dataset Generation

The dataset that was used to create model 1 was provided by [10]. This dataset contains around 4 thousand tweets that were annotated by the research team. These tweets are labeled as either POLIT (political) or NOT (apolitical), as shown below.

```
POLIT     RT @AdamSmithInst Quote of the
week: My political opinions lean more
and more towards Anarchy

NOT     @DeeptiLamba LOL, I like quotes.
Feminist, anti-men quotes.
```

This dataset was put through three processes that would make the text valid to be used for the ML training. The first was putting the data through various filters that would remove symbols, conjunctions, punctuation, links, hashtags, usernames and make the word lowercase. The second process was spell checking which would correct any misspelled words in the text. This was done using the python package "spellchecker"[13]. The last process was lemmatizing the text, which aims to change each word to its headword. Figure 2 below shows the process of lemmatizing a word.

Figure 2: Lemmatizing Process

Figure ?? shows the process of a tweet going through multiple processes to clean it. We can see that the "Preprocess" stage removes the username from the front, lowercases the text, removes punctuation, and removes the URL link at the end. The second stage is spellchecking, and here it corrects the word "problm" to its correct spelling "problem". The final stage is the lemmatizing and here the word "has" was lemmatized to "ha". This process repeats for every tweet from the dataset.

3.1.2 Model Creation

In this subsection, we will discuss how we created the model using Python. We used a python package, scikit-learn, to make the text usable by a ML algorithm. We first need to convert the text into numeric feature vectors, which was done using the CountVectorizer from sklearn's package. Then we normalize the count matrix using TfidfTransformer, which allows us to find the words/tokens that occur most frequently in a dataset.

Finally, we use three different ML algorithms to test against each other on the dataset. These algorithms are the random forest, naive bayes, and the SVM classifiers, as shown in table 1.

From these scores we can see that out of the three algorithms, the random forest and SVM al-

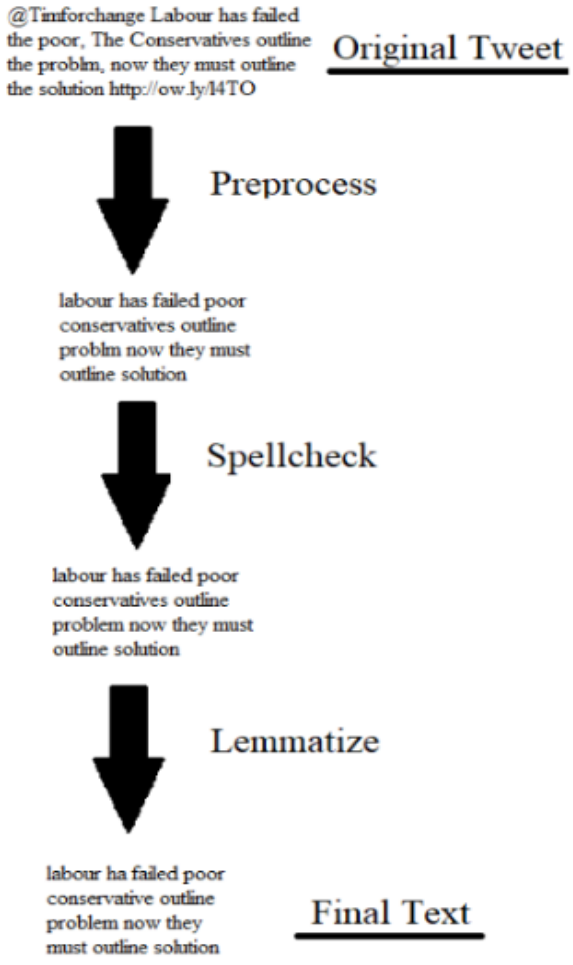

Figure 3: Process of making tweets ready for the ML algorithm

gorithms gave the best scores with the random forest just barely in the lead. We used the random forest model to filter out non-political tweets from Model 2's dataset.

3.2 Model 2

3.2.1 Dataset Generation

The dataset we generated was done using Twitter's API and is composed of two parts. The first is the data for the left-leaning tweets, and the second is the data for the right-leaning tweets. We first selected left and right-leaning politicians as the users to get the data from. Then we analyzed the people who like, retweet, or interact with the tweet and selected the people who are active on Twitter politically. For example, in figure 5 we can see a tweet's likes and the people who interacted with it. We decided to do this because people who often like or retweet a tweet often agree with the original poster. Hence if we wanted to collect tweets from right-leaning users, it is best to start at right-leaning politicians and then see who liked/retweeted the tweet. Once we get that information, we can filter out the non-active users, and finally, we have a list of right-leaning Twitter users on whom we can collect data on.

The same process was done for left-leaning users, and once we collected around 120 users in total, we used the Twitter API to collect likes/tweets from the users. The process for tweet collection was run on google collaborate and took around a couple of hours due to the rate limitations of the Twitter API.

Once the user list was exhausted of tweets that we could collect, our dataset was filled with around 10 thousand tweets. This dataset is split into around 6 thousand tweets from left-leaning users and around 4 thousand tweets from right-leaning users. However, a majority of these tweets are non-political and so model 1 was used to filter out the political tweets from the non-political ones. After this process was finished, we were left with around 3 thousand tweets, or around 1.9 thousand left-leaning tweets and 1.2 thousand right-leaning tweets. This was also after removing duplicate tweets from the database.

Finally, we performed the process of cleaning the tweets using the same method as described for model 1.

3.2.2 Model Creation

This process was similar to the process done for model 1. We split the dataset into 80% training and 20% testing. We used the same three algorithms and tested the results between each other as shown in table 2.

Table 1: Average scores for each ML algorithm Model 1

Algorithm	precision	recall	f1-score	Avg
Random Forest	0.91	0.90	0.90	0.90
Naive Bayes	0.86	0.87	0.86	0.87
SVM	0.89	0.89	0.89	0.89

Table 2: Average scores for each ML algorithm Model 2

Algorithm	precision	recall	f1-score	Avg
Random Forest	0.87	0.87	0.87	0.87
Naive Bayes	0.84	0.82	0.81	0.82
SVM	0.89	0.89	0.89	0.89

The results for random forest and SVM are again very close but this time SVM has the higher

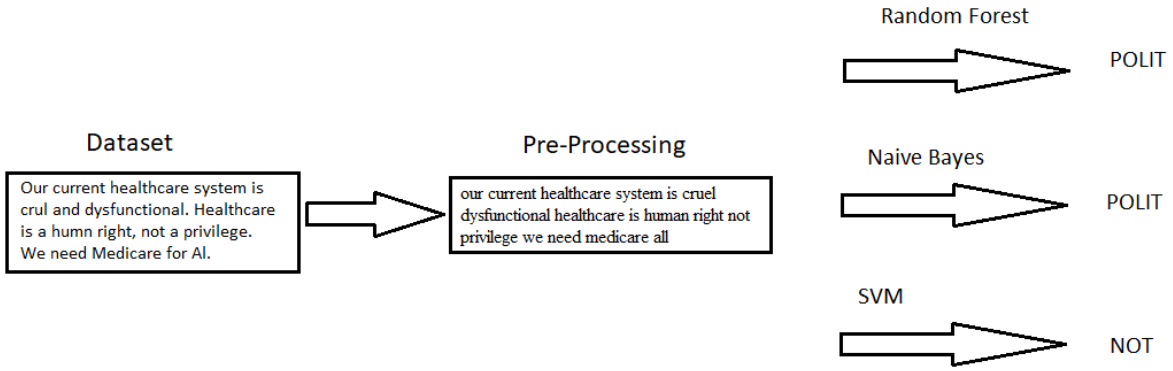

Figure 4: Illustration of the differences between the algorithms in Model 1

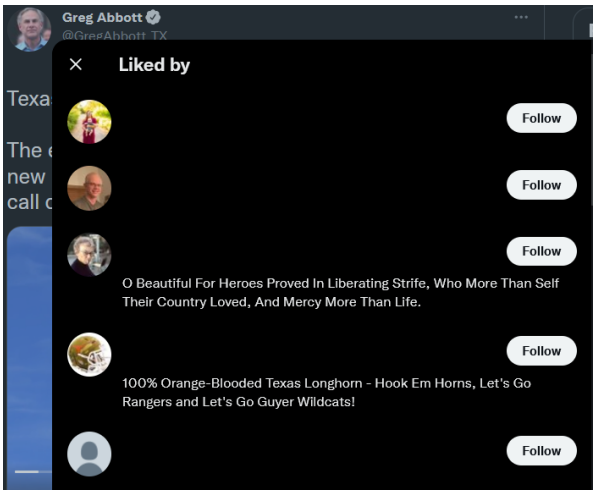

Figure 5: Likes from Greg Abbott's (TX governor) tweet

accuracy score.

3.2.3 Bag of Words

In this subsection we will analyze the word usage and how frequently it is being used in its respected political leaning. Each Bag of Words graph was generated using the same dataset used in model 2. Any word that is not a noun was removed from the BoW model/graph.

In figure 6 we can see the defining words that a left leaning person often says. For example, "trans people" which are the top two words are often a talking point that left leaning people are very vocal about. Similarly "abortion" is 6th most used word, most likely given the fact of the Roe v Wade event that was taking place during this research time period. There is also "debt" and "student" which refers to "student debt" and how left leaning users often call for the cancellation of student debt

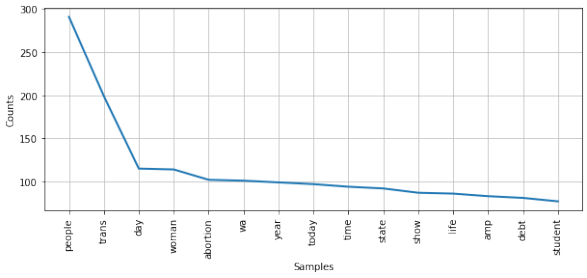

Figure 6: Bag of Words: Frequency of Words in Left Leaning Tweets

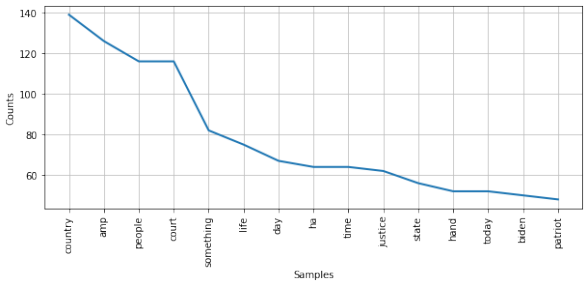

Figure 7: Bag of Words: Frequency of Words in Right Leaning Tweets

In figure 7 we can see the defining words that a right leaning person often says. For example, "country", "justice", "patriot" are words that often describe someone who is right leaning. The word "count" may be in reference to the phrase "stop the count" which is said most often by right leaning people.

4 Results

In this section we will discuss the results and what they mean for our research. Specifically in the resulting dataset, how we used it, how we trained

the models, what we noticed from our dataset, and how we will evaluate our results.

4.1 Dataset

Our final dataset used to train and test model 2 was around 3.1 thousand tweets. Here we had 62% of the tweets being left leaning and the right leaning tweets being 38% of the total tweets. Below is an example of a left leaning tweet from the dataset. Note this is not the exact tweet and is the result of the preprocess we talked about in Section 3

```
"dear please cancel student debt sincerely
 43 million american"
```

A right leaning tweet is shown below as well.

```
"so we maga people are most extreme
 politicalorganization american history
 according corrupt biden yes we are
 extreme follow rule love our
 country mainly one nation under god"
```

However this dataset is not perfect as model 1's accuracy was around 90% this means that around 10% of the tweets should be non political. We can see this in the following example which is a tweet that was labeled left leaning.

```
"best way spend got sunday is with bruce"
```

Or a right leaning tweet such as the one below.

```
"sending prayer big hug both you have
 safe drive"
```

We can see that the top two tweets have nothing to do with politics, however, the algorithm may have assumed the bottom most tweet was political due to the word "prayer" which is something we noticed right leaning users say a lot, along with "amen" or "god". This transitions us to our next subsection "Building the Knowledge"

4.2 Building the Knowledge

Here we will talk about how we patterns in our data we recognized and how we adapted to it. For example, in our first dataset before we tuned the search parameters, our dataset included a noticeable amount of religious and far right wing tweets. This was most likely because the users we collected the data from were part of social circles that shared these kinds of tweets often. To circumvent this we had to collect data from less right leaning politicians and the users who interacted with them a lot. An example of a less extreme religious

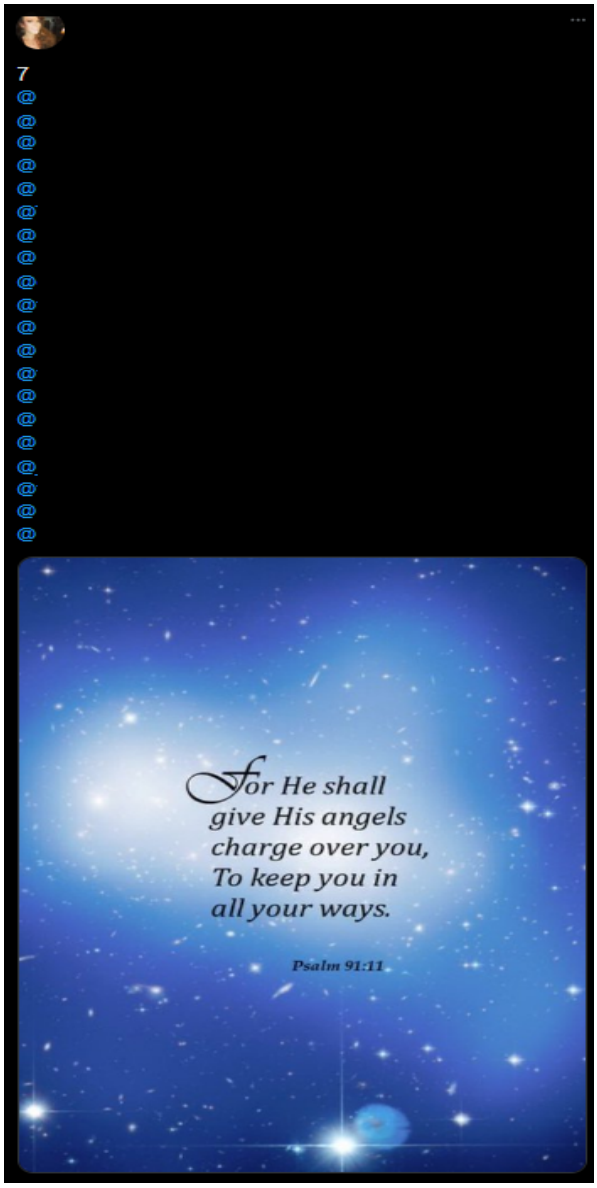

Figure 8: Example of a tweets often shared by far right leaning groups

tweet that is often shared in these groups is shown in figure 8

Each line of the symbol "@" is a different user and is censored for the protection of the users. The point of these tweets seem to be to expand the right leaning social group. While it is not clear whether these accounts are bots or real people, they had a strong influence in the dataset because of the tweets they liked/interacted with. Our original graphs for the BoW models contained more far more religious words like "god" or "amen".

This type of associations is interesting to find and is not something we noticed in left leaning users mostly. Nevertheless for our dataset to generate right wing political tweets properly we had

to find less extreme users that often follow less extreme right wing politicians.

A problem we encountered for left wing users is limiting how far left we wanted to go. For the scope of this research project we wanted to limit ourselves to the political spectrum in the United States. Thus we had to find liberal or socialist users at the very least who were not very extremely left. This was done so that the dataset would be more consistent in finding left leaning tweets successfully. For example, using users who tweet/interact with far left leaning politicians like AOC or Bernie Sanders was not something we used.

```
classifier = RandomForestClassifier(n_estimators=1000, random_state=0)
```

Figure 9: Parameters used for Random Forest Classification

4.3 Training Details

In this subsection we will talk about the details and parameters for each algorithm we used when training them on the datasets. To test fairly all three algorithms we chose to use the same testing and training dataset on all three of them.

4.3.1 Random Forest Classification

For the parameters we used 1 thousand n_estimators. This number picked because of the articles and papers we read and how around this number was good for textual classification. This is shown in figure 9 To train this model we did a split of 80% training and 20% testing

4.3.2 Naive Bayes

For this algorithm we did not customize any parameters and we also had the same split of 80% training and 20% testing. This is shown in figure 10

```
classifier = MultinomialNB()
```

Figure 10: Parameters used for Naive Bayes

4.3.3 SVM

For this algorithm we chose to go with a linear function and C or penalty parameter of 1 since it seemed to give the best result. The parameters are shown in figure 11

```
classifier = svm.SVC(C=1.0, kernel='linear',
 degree=3, gamma='auto')
```

Figure 11: Parameters used for Support Vector Machine

4.4 Evaluation Details

From our evaluations we chose to use the random tree classifier for model 1 due to its higher accuracy score. For model 2 we chose to use a linear SVM because of its high accuracy score.

5 Conclusion

In conclusion, we built two models. The first is a model to predict if a given tweet is political or not. The second model is to predict the political leaning of the tweet. We used three different algorithms and compared them to find which one was the best. For model 1, we chose to use the random tree classifier algorithm, and for model 2, we chose the SVM. We hope this research provides a stepping stone for other researchers in their attempts to collect and analyze political tweets.

References

[1] Ruogu Kang, Stephanie Brown, and Sara Kiesler. Why do people seek anonymity on the internet? informing policy and design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 2657–2666, New York, NY, USA, 2013. Association for Computing Machinery.

[2] Nugroho Dwi Prasetyo and Claudia Hauff. Twitter-based election prediction in the developing world. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, pages 149–158, 2015.

[3] Hao Wang, Doğan Can, Abe Kazemzadeh, François Bar, and Shrikanth Narayanan. A system for real-time twitter sentiment analysis of 2012 us presidential election cycle. In *Proceedings of the ACL 2012 system demonstrations*, pages 115–120, 2012.

[4] Wikipedia Contributors. Twitter — Wikipedia,, the free encyclopedia, 2019. [Online; accessed 27-Octobar-2020].

[5] Marco Pota, Mirko Ventura, Hamido Fujita, and Massimo Esposito. Multilingual evaluation of pre-processing for bert-based senti-

ment analysis of tweets. *Expert Systems with Applications*, 181:115119, 2021.

[6] P Ekman. Universals and cultural differences in facial expressions of emotion in nebraska symposium on emotion and motivation, 1971 (ed. cole j) 207–283, 1972.

[7] Bahadorreza Ofoghi, Meghan Mann, and Karin Verspoor. Towards early discovery of salient health threats: A social media emotion classification technique. In *biocomputing 2016: proceedings of the Pacific symposium*, pages 504–515. World Scientific, 2016.

[8] Dharini Ramachandran and R Parvathi. Analysis of twitter specific preprocessing technique for tweets. *Procedia Computer Science*, 165:245–251, 2019.

[9] Farhan Hassan Khan, Saba Bashir, and Usman Qamar. Tom: Twitter opinion mining framework using hybrid classification scheme. *Decision support systems*, 57:245–257, 2014.

[10] Micol Marchetti-Bowick and Nathanael Chambers. Learning for microblogs with distant supervision: Political forecasting with twitter. Technical report, MICROSOFT CORP SAN FRANCISCO CA, 2012.

[11] Felix Biessmann, Pola Lehmann, Daniel Kirsch, and Sebastian Schelter. Predicting political party affiliation from text. *PolText 2016*, 14:14, 2016.

[12] Daniel Xiaodan Zhou, Paul Resnick, and Qiaozhu Mei. Classifying the political leaning of news articles and users from user votes. In *Proceedings of the international AAAI conference on web and social media*, volume 5, pages 417–424, 2011.

[13] Tyler Barrus. pyspellchecker. [Online; accessed 11-May-2022].