Accessing Public Attitude Via Twitter Sentiment Analysis

Michael O'Malley
University of Notre Dame
Notre Dame, IN
momalle3@nd.edu

Michael Burke
University of Notre Dame
Notre Dame, IN
mburke18@nd.edu

ABSTRACT

Social media sites such as twitter and facebook are becoming increasingly influential in the way that our society operate. Companies can use these sites to market and advertise, but they can also fall victim to a scandal. Political activists from all over the political spectrum can offer sentiments and thoughts on recents events. Normal people (people that don't have celebrity status) can offer their opinions and find that one of their tweets has gone viral. In a nutshell, the wealth of information that these sites provide is a powerful tool in today's atmosphere. Being able to compile large amounts of blurbs specific to an event and evaluate the sentiments of these blurbs can paint a better picture of the overall public feeling towards the event. One can search beyond highly visible celebrities and news outlets and reach a larger pool of opinions. Our project aims to analyze the sentiments of tweets from twitter that regard specific events. This will allow us to build a better picture of public opinion and general sentiment towards an event.

1 INTRODUCTION

Our project aims to properly predict the positive or negative language of a user's tweets that are specific to an event or hashtag trending on twitter. Since every trending item has its own language, it's important to learn compile these unique sentiments and learn from them. With the advancements of websites' ability to filter certain kind of content based on a user's preferences, the ability to categorize a tweet based on keywords and phrases is very important. Since the website Twitter remains very popular, we find this challenge interesting and very applicable for today. Ideally, our approach would be able to segment a user's messages into good, or positive, and bad, or negative, groups.

2 DATA SOURCES

Luckily, in the age of social media, there is an abundance of highly opinionated sentiments floating around on the internet that allow for compilation and analysis. More specifically, this data can be sorted by keywords and labels, which can help us to follow specific events and trending topics.

2.1 TWITTER API

Our most important resource will be Twitter data. Twitter allows users to register to use their API, tweepy, to periodically scrape and culminate their data. Once registered, tweepy has two tweet collection functionalities: search and streaming. Search allows you to input specific keywords and a max number of tweets, then returns tweets that matched your search. This tends to output more visible and popular tweets, which has pros and cons. The streaming functionality similarly allows you to filter tweets by keyword, however instead of immediately returning results it waits for new tweets that fit the parameters and continuously returns them to the user on an open channel. The streaming only stops on Keyboard interrupt or when the Twitter API limits your data. Both of these functionalities are useful, and both will be used to compile databases to use with our sentiment analysis.

2.2 SENTIMENT ANALYSIS TRAINING DATA

Twitter will provide the data that we can apply our model to, but we first have to build an accurate model. This will require training and testing data. Luckily, sentiment analysis has been worked on previously by several sources, and has plenty of training data available. Cornell has done extensive research on the matter, and offers databases from places like reddit and imdb. Additionally, the MPQA Opinion Corpus has compiled databases with graded sentiments and opinions. In terms of specific twitter sentiment, Niek Sanders has compiled an enormous corpus of information that can be used to train general models. In order to implement the learning part of our algorithm, i.e. the part where the model learns the specific language of a given trending item, we'll have to manually grade many of our own tweets. Regardless, we have plenty of places to go for training data that will help us construct an effective model specifically for sentiment analysis.

3 RELATED WORK

Over the past decade, interest in dissecting meaningful trends and information from social media has increased greatly due to the large number of users worldwide. Individuals use the Twitter platform to share personal opinions, news updates, financial advice, and intellectual conversation. Researchers look to use Twitter as a way to make predictions about stock market movements [1], government elections [2], and potential crimes [3]. Furthermore, several papers have worked towards analyzing the sentiment of tweets. Researchers examined the way certain popular Twitter users are able to manipulate the sentiment surrounding certain social issues and events in a positive or negative way [4]. We hope to learn from these works' strategies to correctly predict the public sentiment relating current events, and we view this goal as very important due to the polarizing climate we have today.

4 METHODS

There are many steps in our analysis process: retrieving data, cleaning it for use, training a model, applying that model, evaluating results, and several others. Naturally, this includes a variety of methods that practice clean and efficient data processing. Each step had its own methods.

4.1 DATA COLLECTION

As mentioned before, the data collection comes in two parts: training data and testing data. Obtaining the training was as easy as downloading Niek Sanders' Twitter Sentiment Corpus. His database contains over 1.5 million tweets, each with an ID, a sentiment score, a source of the score, and the text of the tweet. This is compiled in a CSV, an easy file format for the python pandas library to read in. Additionally, the scores are binary, the value 1 representing a positive and the value 0 representing a negative tweet. As helpful as this is, it is an incomplete set. Our goal is to train a model based on the language of a specific event. So, on top of this data, which can be used to train our general model to be built upon, we also constantly collect data regarding specific events. This is the data that we can split into training and testing instances. Our twitter scraper takes a single phrase as an input, then opens up a Tweepy stream listener that finds all instances of that keyword (whether it be in normal text or part of a hashtag). When the scraper gets a hit, it returns a massive JSON file with tons of tweet information, most of which is noise. This is where the data cleaning begins.

4.2 DATA CLEANING, INTEGRATION

Tweepy finds plenty of useful information, like the text and timestamp of relevant tweets, but it also finds plenty of useless data, like contributors and entities. The first step towards properly analyzing this data is eliminating the noise. Right of the bat, our scraper receives the Tweepy JSON and only records four things: The user id number, the text of the tweet, the timestamp of the tweet, and the location of the tweet (which is often not given). This process is visualized in *Figure 1* on the next page. On top of this initial screening, we have a tweet cleaning script that brings the data into a pandas dataframe (where it is easier to work with), replaces each gap in location with a None variable, throws out rows that have empty or mostly empty text, and repackages the data into a csv rather than a text file. This cleaning integrates the data nicely into existing data and prepares for the next step in the analysis process.

4.3 MODEL TRAINING

Our model is broken up into two steps, both of which include different technologies and classifiers. The first step involves training our model based on the enormous sentiment analysis database we found. Using a popular sentiment analysis library called textBlob, we are able to train to different models with

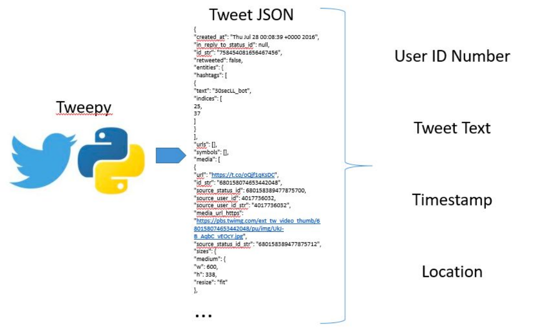

Figure 1

our data: A Naive Bayes Classifier and a Decision Tree Classifier. Training two different models allows us to get slightly different evaluation metrics, helping us to build an ensemble that is more effective than a single classifier. All of this is exhibited in *Figure 2*. After training our models with this data, we move on to the next step: training a neural network with the event-specific twitter data. This process is a little more complicated, but it adds the layer of learning that will differentiate our classifier from typical sentiment analysis algorithms. Using the accumulated data and the newly trained model, our training script builds a new database with the user id number, the text of the tweet, the sentiment score according to our newly trained models, the timestamp, and the location if available. These are all included for specific reasons: the user id number can be recognized as a typically negative or positive contributor, the text is included for obvious reasons, the new score represents an analysis based on previous tweets rather than general language, the timestamp could reflect a development in the trend after a real life occurrence, and the location could hint at an area that typically sways one way or another. All of this data is fed through a few new classifiers. The most important one is a neural net built to identify foreign language and weigh it. The net keeps track of language specific to a hashtag by adding any unknown language to a dict (for example, a term like deflategate would not turn up in a normal dictionary, but it was incredibly popular during the January 2015 playoffs). The dict contains the new words and, more importantly, then weights associated with them. As the data is updated, the weight of a word is updated based on its appearances and the sentiment score associated with it. The words and their weights are then added back into the analysis to further train the neural net to recognize the negative or positive nature of these event-specific words. On top the neural net, we also implement another ensemble of models to analyse the remaining data: the user id number, the timestamp, and the location. These pieces of information represent possible changes in the sentiment of a tweet. By including them in a new ensemble classifiers, combining those results with the neural net results, and producing results based on all of the available data, we produce a much more supported and sound output. The second step of this process is exhibited in *Figure 3.*

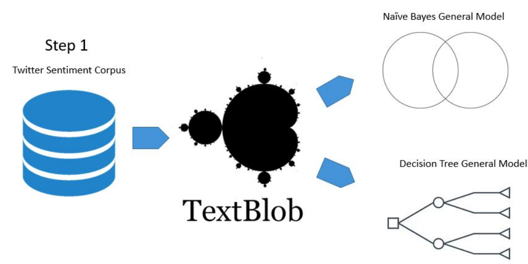

Figure 2

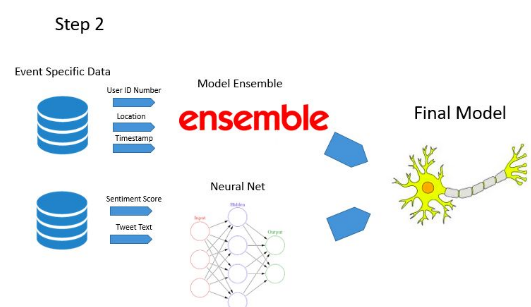

Figure 3

4.4 MODEL APPLICATION

Considering how convaluated our training process is, the application process is much more simple. After training the models explaining in section 4.3, testing is easy, yet painstaking. We have to go through a significant amount of tweets for each set of data, read the available text, and classify each tweet individually. While time-consuming, there are no available options for classifying such new tweets, given the recent nature of our data. After doing this for a significant chunk of data, we now have a test set. Testing our models is as simple as running this new data through the models generated in our second training step.

4.5 MODEL EVALUATION

There are plenty of options for model evaluation when it comes to machine learning. We chose to use both conventional methods and a clustering method that can offer insights into tweet similarities. As for the traditional metrics, we calculate the accuracy, recall, precision, and F1 score. These basic metrics paint a good idea of how successful our algorithm is when working on the test data. It's a good idea to look outside the box, however. In addition to these metrics, we decided to cluster the tweets based on the similarity of their language. We did this by calculating the Jaccard score between two tweets. This score is the difference between two sets of words: basically, it calculates the similarity between two sentences. By calculating the Jaccard scores of each tweet, we're able to cluster them according to their closeness to other tweets. This gives us a good idea of which tweets belong where. Tweets with similar language often say similar things, and negative and positive tweets should end in separate clusters. By mapping these clusters and comparing them to the classifications form our model. We generate a good idea as to where thy'rer supposed to be versus where they are.

5 RESULTS

Currently, our results don't offer us much. There are many bugs in our implementation, leading to inconsistencies in evaluation metrics and clustering comparison results. In other words, before we can properly evaluate the results of our model, we need to finish the implementation correctly.

6 CONCLUSIONS

Sentiment analysis is very tricky. A single word can change the entire sentiment of a sentence, making it very difficult to represent in numerical form. The way our sentiment analysis differs is that we build upon previously established methods of analysis and continue to learn from them. Our attempt at event-specific sentiment analysis is unique in the idea that we can learn the new language associated with an event and adapt to it, rather than just relying on analysis performed in the past. This addition to the extensive work done in sentiment analysis can help to fit the constantly changing political and social landscape that the modern age has built. With the abundance of new and unique data, our model attempts to understand the fast-paced nature of social media trends.

7 REFERENCES

[1] Sentiment Analysis of Twitter Data for Predicting Stock Market Movements.
https://arxiv.org/pdf/1610.09225.pdf
Datasets, www.cs.cornell.edu/home/llee/data/.

[2] Bayesian Twitter-based Prediction on 2016 US Presidential Election
https://arxiv.org/pdf/1611.00440.pdf

[3] Predicting crime using Twitter and kernel density estimates
https://www.sciencedirect.com/science/article/pii/S0167923614000268

[4] Sentiment analysis of twitter audiences, Measuring the positive or negative influence of popular twitterers
http://onlinelibrary.wiley.com/doi/10.1002/asi.22768/full

[5] Sentiment analysis of Twitter Data
https://dl.acm.org/citation.cfm?id=2021114

[6] Coooolll: A Deep Learning System for Twitter Sentiment Classification
http://www.aclweb.org/anthology/S14-2033

"Getting Started¶." Getting Started - Tweepy 3.5.0 Documentation, docs.tweepy.org/en/v3.5.0/getting_started.html.

MPQA Opinion Corpus | MPQA, mpqa.cs.pitt.edu/corpora/mpqa_corpus/.