Deconstructing the 2016 US Presidential Election With Twitter

Shirish Dhar, Boris Lo, Harman Shah Singh, Parv Sondhi

December 12, 2016

1 Introduction

With everyone walking around with a smartphone, we are living in a world where various forms of social media such as Facebook, Twitter and Instagram form an integral part of everyone's life. Be it sharing photos from a latest trip or inquiring about a particular product's performance online, social media serves everything on a platter and the user can choose which feature to use when. This has also led to the emergence of a very interesting political aspect of social media - using Tweets as a medium of sharing political views, whether support, opposition, agitation or disappointment, related to a particular incident or party. This formed the basis of our project. The original intent of the project was to analyze the tweets obtained from 2016 Presidential Election Data Hackathon [1] to understand if the semantics and sentiments of these tweets could be used to predict/ understand the results of the US presidential elections 2016. In a broader sense, our project aimed to answer the question "Do people tweet the same way they vote?"

2 Data

The dataset is 40000 tweets from the first week of October with half of them categorized as tweets about Clinton and half of them categorized as tweets about Trump taken from the 2016 Presidential Election Data Hackathon. In addition to the tweet text the data also included a user id, id number, number of followers, number of retweets, user id the tweet is replying to, number of favorites, number of friends, and roughly a third of them are geotagged. For the purposes of this project we only used the text and location information. Some sample tweets with their location data are shown in Table 1, as shown the geotag data is not standardized.

Table 1: Samples of tweets from the original data set before any preprocessing. The location assoicated with these tweets, if they are present, are not standardized; some have the city, others only the state, and still others just the continent or country.

Original Text	Location
No privatization of SS; I've paid in, I want mine! Yes Clinton/Kaine; No Trump/Pence-not presidential!	Davenport, IA
RT @KevinMKruse: Pence claimed Clinton's plans would raise the debt more than Trump's. No. https://t.co/ws5ThkqnZ4	Michigan
RT @USAforTrump2016: I've got a question Tim...Should Hillary Clinton face consequences for mishandling classified information? #VPDebate	North America

3 Algorithm

Broadly speaking, the tweets can be classified into five categories:

- Pro-Clinton
- Pro-Trump
- Anti-Clinton
- Anti-Trump
- Neutral

As can be very well perceived, given the pre-election scenario where Trump and Clinton were the primary contenders for the presidential position, any tweet that is Anti-Clinton can be abstracted as Pro-Trump. Similarly, an Anti-Trump tweet can be considered as Pro-Clinton. In addition, since the tweets are taken to be about one of the candidates the number of truly neutral tweets are assumed to be minimal. Thus, we can reduce the above categorization into Pro-Clinton and Pro-Trump.

3.1 Preprocessing

Given the structure of the tweets obtained as mentioned above, it was important for us to spend effort and time to clean up and pre-process the tweets to obtain text in format that could be used for further analysis. The data associated with each tweet was not standardized, and this required us to follow certain steps to standardize the data. Our pre-processing consisted of the following steps:

1. **Tokenization and Tagging**
   We started by tokenizing our tweets to gather individual tokens within the text to clean up. We set up a regular expression parser to remove all urls from our tweets to help filter out only text data that was of a natural language format. We then removed all instances of usernames (eg. @evshear) from the tweet. This set up our tweet to have only text and hashtags remaining within the tweet, along with the unstructured location data stored separately.

2. **Hashtag Parsing**
   We filtered out all hashtags from the tweets by tagging relevant tokens as Hashtags. These hashtags were segregated and added to a separate column and removed from the tweets. Each hashtag was parsed and stored as an association to the tweet to which it belonged as part of the final data frame.

3. **Location Extraction**
   With each tweet we saw that the location data was not stored in any standard format and it was important to convert each of these to a state abbreviation which could be used to easily cluster tweets by states. For each location we parsed through the location text using a regular expression grammar to convert it to the relevant state abbreviation. If no US state can be identified then it is left as None. For this, Washington DC is taken as a separate state.

Table 2 shows the initial state of two sample tweets and their preprocessed forms after. Table 3 shows that after preprocessing the number of tweets for each candidate in each state.

Table 2: Some sample tweets and their preprocessed results. The preprocessing stage takes the original location and generates the state the tweet is from. We take Washington DC as a separate state.

Original Tweet	Original Location	Preprocessed Tweet	State	Hashtag
RT @TonyFratto: The funniest parts of both Trump and Pence are when they flat-out deny actual Trump quotes. #VPDebate	Washington, DC	The funniest parts of both Trump and Pence are when they flat-out deny actual Trump quotes	DC	['#VPDebate']
#PrivateEquity Wagers Campaign Cash on Clinton, Senate Races https://t.co/F1roapqAhw @dawnmlim @LCooperReports	United States	Wagers Campaign Cash on Clinton Senate Races	–	['#PrivateEquity']

Table 3: Number of tweets per state. The tuples given are (state, number of clinton tweets, number of trump tweets)

(AK, 11, 8)	(AL, 88, 100)	(AR, 72, 32)	(AZ, 113, 136)	(CA, 770, 894)
(CO, 144, 111)	(CT, 83, 63)	(DC, 204, 193)	(DE, 16, 11)	(FL, 622, 510)
(GA, 250, 281)	(HI, 15, 26)	(IA, 43, 56)	(ID, 28, 16)	(IL, 271, 285)
(IN, 180, 134)	(KS, 54, 51)	(KY, 81, 80)	(LA, 110, 76)	(MA, 188, 231)
(MD, 94, 113)	(ME, 43, 20)	(MI, 175, 154)	(MN, 122, 96)	(MO, 117, 178)
(MS, 32, 29)	(MT, 10, 15)	(NC, 206, 190)	(ND, 1, 4)	(NE, 34, 43)
(NH, 20, 64)	(NJ, 169, 171)	(NM, 14, 33)	(NV, 80, 83)	(NY, 622, 749)
(OH, 319, 226)	(OK, 88, 67)	(OR, 82, 80)	(PA, 308, 270)	(RI, 27, 16)
(SC, 127, 90)	(SD, 1, 3)	(TN, 126, 131)	(TX, 588, 565)	(UT, 33, 55)
(VA, 168, 177)	(VT, 5, 6)	(WA, 199, 236)	(WI, 95, 79)	(WV, 16, 28)
(WY, 5, 11)				

3.2 Statewise Topic Clustering

We started by segregating tweet data for each state and filtering out only the nouns, since topic words are essentially give by just the nouns within the tweets. The results from this step were fed into a TF-IDF Vectorizer, so as to make sure that the words were weighed by their importance within the set of tweets under analysis, and not by how popular those words are in general.

We performed K-means clustering on this data with cluster values of 3, 5, 8, 10, 15 and 20 to come up with a set of clusters that had top words for each cluster and a corresponding set of states. A key observation here was that for all values of k (number of clusters) beyond 5, the semantic relationship between the states in terms of the top trending words was getting lost: the resulting words were too vague to add any value to our analysis. In addition, a value of 3 resulted in too many words falling under each of the three cluster categories to make any distinction between each of the clusters. Hence, we used the cluster size of 5 as our best possible results. The words correspond to the top trending topics and the states give the names of all the states that were talking about those topics.

3.3 Initial Vader Sentiment Analysis

We applied the NLTK implementation of vader sentiment analysis on each preprocessed tweet. Vader sentiment analysis returns four scores per input: a positive, neutral, and negative score on a [0,1] scale and a compound score that is computed separately on a [-1,1] scale where -1 is most negative and 1 is most positive. For this analysis each preprocessed tweet is one input to vader regardless of the original formatting or grammatical structure of the original tweet. Using these scores we interpreted that a tweet is positive for a candidate if the tweet is pre-categorized for that candidate and the scores are positive. Table 4 shows some sample preprocessed tweets with their vader scores. We then used these vader scores to decide which candidate wins a state using various rules and generated an electoral map. The first rule we tried was whoever had a higher average compound score in the state. Another rule we tried was a best-of-three rule where the three metrics are higher average positive, higher average neutral, and lower average negative score. The rule that worked best (gave us the most states correct), is that if the average positive score is higher and the average negative score is lower for a candidate then that candidate wins the state. If this metric is a tie then the candidate with the average higher compound score wins the state. Using this rule, we got a 61% accuracy in terms of predicting who the winner would be in a particular state vs what the actual results were. The result is shown in Fig. 1. However, more alarming was the fact the California, New York, Oregon, and Washingont, four states which are predominantly Democratic, returned a result in Favor of the Republicans (Trump) as shown in Fig. 1. This motivated us to more closely look at what may be the reason for the unexpected results from our analysis. A closer inspection of the tweets revealed that:

1. The classifier wasn't doing a good job at detecting sarcasm.

2. The classifier was misclassifying the tweets that had a complex structure. For instance the tweet "Pence said Hillary is worse than Trump. Definitely not!" was in the Clinton category but with negative sentiment scores thus classified as Anti-Clinton.

3. The classifier misclassified the tweets that had mentions for both Hillary and Trump, owing to the fact that it couldn't judge who the tweet was more negative or positive towards.

Ultimately, we realized vader gives only sentiment of the tweet but not sentiment towards the subject.

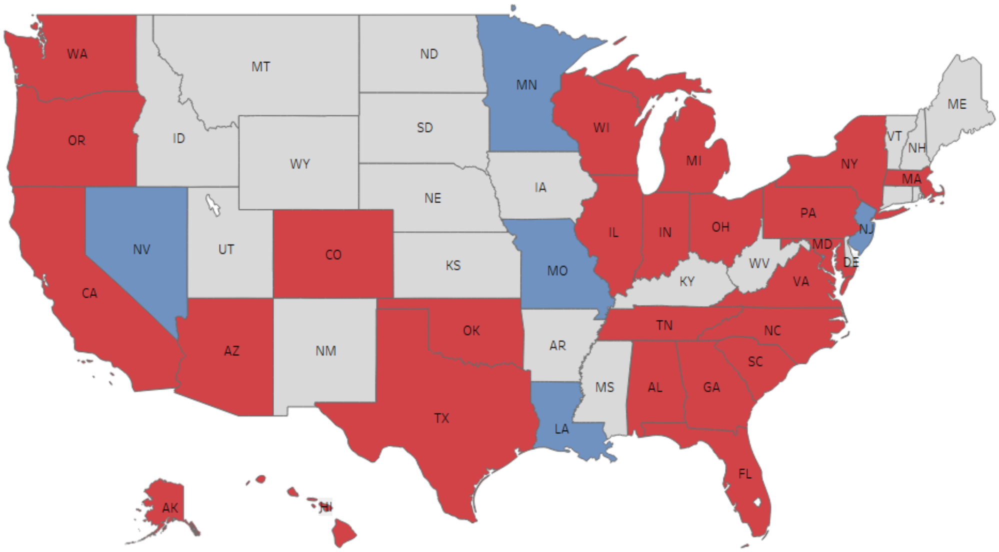

Figure 1: Electoral map based on vader scores on preprocessed tweets. Red states indicate Trump won based on our analysis and blue states indicates Clinton won. Grey states are states where we had less than 150 tweets combined across both candidates.

Table 4: Some sample preprocessed tweets with their vader scores and our interpretation of the tweet based on these scores. In our analysis a tweet that is against a candidate is considered pro the other candidate.

Category	Preprocessed Tweet	Positive	Negative	Neutral	Compound	Interpretation
Clinton	How Hillary Clinton helped to crush democracy in	0.0	0.186	0.814	-0.1531	Anti-Clinton
Clinton	LIVE FACT-CHECK Kaine said the Iran nuclear deal dealt with Iran “without firing a shot”	0.257	0.0	0.743	0.5908	Pro-Clinton
Trump	Kaine on nuclear weapons Trump could be the fool or maniac who triggers a catastrophic event	0.0	0.524	0.476	-0.9022	Anti-Trump
Trump	Kaine dgaf about how he comes off He just wants to keep pounding Trump Pence cares a lot about how he comes off	0.125	0.0	0.875	0.4588	Pro-Trump

3.4 Our Tweet Classifier

3.4.1 Goal and Preparation

After the initial simple vader analysis, we dived deeper into the tweets and aimed for a better classification algorithm. We combined the tweets from the two categories (Clinton and Trump) and decided to work with them together rather than performing analysis on the two groups separately then comparing the result. We took every tweet from CA and NY as well as some from KY and manually classified them as either Pro-Clinton or Pro-Trump. Out of the $\sim$ 1300 tweets from this process only 296 are Pro-Trump. In order to not bias the machine learning model, we randomly sampled 296 Pro-Hillary tweets and this combined 592 tweets served as the data set for our classifier.

3.4.2 Features

For the purposes of describing the features, we identify the Hillary camp as the four proper nouns: (hillary, clinton, kaine, tim) and the Trump camp as the four proper nouns: (trump, donald, pence, mike). In addition, we wanted to focus on the NLP and so we neglected the tweet metadata, as described in 2 that came with the data set.

1. **Name Occurence**
   The first two features are binary features of whether a name in the Hillary camp appeared and a name in the Trump camp appeared. In addition, we incorporated a binary feature for if Bill Clinton appeared in a tweet because we found from looking at the tweets that Bill Clinton is a name very often used to negatively target Hillary without any other names present in the tweet.

2. **Distance between Subject and Negative/Positive Word**
   Looking at the structure of the language of the tweets we realized that there was high usage of certain words of negative and positive sentiment that were being used with respect to the subjects of those tweets. As a result, we decided to calculate the distance between the subject and the closest negative/positive word and use that as features for our model.

   To construct our negative and positive word list, we ran the 40,000 tweets through our code snippet to extract all unigrams and compute the individual sentiment score for them through the Vader sentiment analyzer. For all unigrams that had a positive/negative score of 1/0 were tagged in the positive word list and similarly those that had a positive/negative score of 0/1 were added to the negative word list. In addition, we added the words not, hasnt, cant, didnt, wont, doesnt, and n't into the negative word list. We added n't because the tokenizer we used for this step is the NLTK word tokenizer which split words such as “didn't” as ['did', 'n't'].

   This combination was used to compute 4 features for each tweet:

(a) Distance between positive words and subjects for the Hillary camp

(b) Distance between positive words and subjects for the Trump camp

(c) Distance between negative words and subjects for the Hillary camp

(d) Distance between negative words and subjects for the Hillary camp

For each tweet, we then calculate the minimum distance between any word appearing in either of the camps and a negative/positive word if one exists in the tweet. If no negative/positive word is found in the tweet or the tweet does not contain any word from either camp, we assign the feature a value of maximum distance (=20).

3. **N-gram Vincity Sentiment**

For each tweet with multiple subjects being mentioned due to cross-talk we realised that certain tweets had multiple opposing sentiments occurring within a single tweet which were skewing vader sentiment results when tagging as Pro-Trump or Pro-Hillary. For the purposes of this feature we identified n-gram chunks around the important subjects for both Hillary and Trump camps within the tweet and calculated the compound score from Vader for these individual chunks. Once calculated, we summed over all the individual compound scores to obtain final 2 values for the Tweet.

These individual scores helped establish a better sense of positive or negative sentiment around the particular subjects in question for the tweet. For example, lets take a look at the below tweet which was tagged as a pro trump tweet during the initial vader sentiment analysis:

Tweet - Do you want a You're Hired president in Hillary Clinton or do you want a You're Fired president in Donald Trump Computed Score:
'compound': -0.4588, 'neu': 0.721, 'neg': 0.162, 'pos': 0.117

Through our N-gram vicinity sentiment feature, we get two n-gram chunks from the tweet based on the subject:

Chunk 1 - You're Hired president in Hillary Clinton or do you want
Computed Score:
'compound': 0.0772, 'neu': 0.874, 'neg': 0.0, 'pos': 0.126

Chunk 2 - do you want a You're Fired president in Donald Trump
Computed Score:
'compound': -0.5106, 'neu': 0.588, 'neg': 0.303, 'pos': 0.109

As a result we see that for this tweet we get a Hillary_sentiment value of 0.0772 while Trump_sentiment value of -0.5106. As we see above there is a much clearer negative sentiment towards Trump and a positive sentiment towards Hillary.

4. **Topics**
   Looking at the tweets and having followed the election it was immediately clear that there were several broad topics that people often tweeted about during the presidental campaign. For example, Clinton's emails or Trump's border wall are likely good indications of whether a tweet favors one candidate or the other. Table 5 shows eight topics and words assoicated with these topics. This corresponds to eight binary features where 1 indicates that the tweet contains a word associated with that topic. The first six topics were taken from the `ReportTwitterAnalysisofPresidentialCandidates.pdf` in [1].

Table 5: Topic feature and words associated with them.

Topic	Words
email	emails, email, crookedhillary, prison, crooked
russia	putin, russia, crimea, vladmir, ukraine, russian
race	white, black, racist, race
immigration	borders, border, wall, mexico, illegal, immigrants, immigration, trafficking
trust	factcheck, fact, factchecking, bigleaguetruth, trust, politifact, trustworthiness, lies, lie, truth, liar
sex	sex, sexual, scandal, rape, assault, transgression, transgressions
female	she, her, she's, herself
male	he, him, his, himself, he's

3.4.3 Features We Didn't Use

As part of our process we also worked on a couple of features that ultimately didnt make it to the final model:

1. **Tweet Hashtags**
   For each tweet we segregated the hashtags and parsed them to get individual hashtags for each tweet. We saw that the hashtags gave us relevant information for the sentiments towards the individual subjects. We calculated the frequency distribution of the hashtags and computed the 100 most common hashtags for the dataset. However, we see that there were only 1237 unique hashtags. This was a relatively low number as compared to our tweet dataset and as a result it was dropped as a feature from our model.

   We did see some interesting results during our analysis, Table 6 shows some of the most common hashtags were, but they are relatively low numbers to be considered appropriate to train our model.

Table 6: Most common hashtags and their occurrences.

Hashtag	Occurrence
#vpdebate	352
#trump	265
#hillary	113
#maga	79
#clinton	69
#pence	63
#kaine	56
#imwithher	42
#neverhillary	34
#nevertrump	29

2. **Part of Speech Usage**

During our analysis, we decided to understand the usage of different parts of speech used by tweeters when tweeting pro trump or pro hillary. After cleaning the tweets we passed them through our trained backoff tagger to tag the parts of speech in our tweet. We then analyzed the parts of speech amongst the tweets for trump and hillary to understand the usage of such words.

We then looked at the most common adjectives, nouns and verbs for each of the camps to understand the differences in the occurrences. Table 7 shows the results. For verbs and nouns we did not see any drastic differences in the commonly occurring terms, however, that was not the case with adjectives. We saw there were certain terms that were unique to both camps and as the next step we created a bag of words which contained the adjectives that were unique to each set.

Table 7: Average number of parts of speech for the two camps.

Camp	adjectives/tweet	nouns/tweet	verbs/tweet
Trump	0.75	5.926	1.8405
Hillary	0.694	6.313	2.171

Once this bag of words was created, we engineered a feature to capture the number of occurrences of such terms in the tweets and calculate the count within each camp.

However, this feature did not help boost our model score, and as a result we ended up dropping the feature from our model.

3.4.4 Machine Learning Model

From scikit-learn we used their multilayer perceptron classifier (MLPClassifier). We used bagging on five MLPClassifiers with all the default setting (each has a different random initial state) and this served as our classifier. We used a 80/20 training/development set split and ran the classifier many times until a good development score was reached. The resultant classifier is the one used to analyze the tweets from all the other states. We considered using other models such as Logistic Regression and Decision Tree but both yielded worse development set accuracy than the bagged MLP.

4 Implementation

Resources and software used:

1. Software and libraries: scikit-learn [2] for the machine learning models and clustering, NLTK [3] for the stemmer, tagger, tokenizer, and their vader implementation, flask for web interface, and pandas and numpy

2. Data: `https://github.com/WiMLDS/election-data-hackathon/tree/master/candidate-tweets-oct-2016`

3. Map Visualizations: tableau

The source code is divided into five IPython notebooks:

1. `Pre-Processing.ipynb`
   Contains all the preprocessing as described in 3.1. It generates the preprocessed tweets as used for all of our analysis.

2. `Clustering.ipynb`
   Contains the clustering as described in 3.2.

3. `InitialVader.ipynb`
   Contains the initial vader analysis as described in 3.3.

4. `MachineLearning.ipynb`
   Contains the features and machine learning. The first part of the notebook contains separate functions that generate the various features, the function `extractFeatures` takes in a pandas dataframe with the preprocessed tweets, calls all the feature generating functions and outputs a features matrix. The latter part of the notebook trains a machine learning model and runs it.

5. `UnusedFeatures.ipynb`
   Contains the two unused features as described in 3.4.3.

4.1 Web Interface

Once we developed the algorithm we decided to user test this model by creating a flask application that allows users to enter any tweet that they would have tweeted out during the presidential campaign. Fig. 2 shows what the interface looks like.

As part of this web application we predict whether the tweet would be a pro trump tweet or a pro hillary tweet. Users have the ability to see how the sentiment towards individual subjects help push the overall sentiment of the tweet.

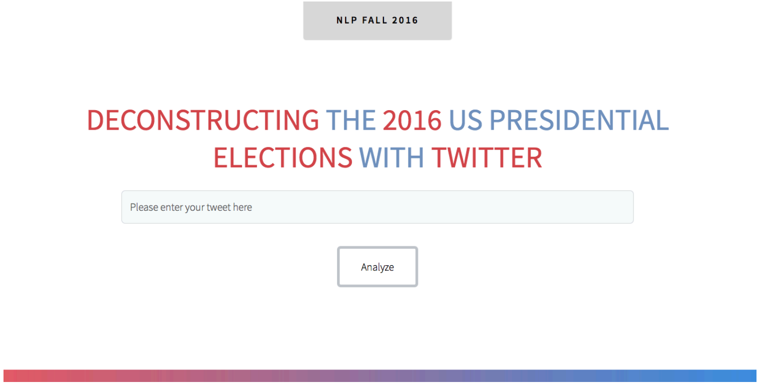
NLP FALL 2016
DECONSTRUCTING THE 2016 US PRESIDENTIAL ELECTIONS WITH TWITTER
Please enter your tweet here
Analyze

Figure 2: Homepage of the webapp.

The web interface can be downloaded from `https://github.com/parvsondhi/NLPFinal2016` because the image files are too large in size to submit.

5 Results and Evaluation

As part of our project we aimed to realize that people tweet the same way they vote. We decided to achieve this by being able to recreate the electoral map and 2016 election results through our analysis of twitter sentiments of voters against the presidential candidates.

Our preliminary result through the vader sentiment analysis gave us a **61%** match with the election results (refer to Fig. 1). The results were unsatisfactory and we decided to do some analysis to understand the anomalies that we got during the first phase. As mentioned above, the mismatch in California and New York pushed us to develop another Subject based Tweet Classifier using the above algorithm to help boost our match with the electoral map.

Using our Tweet Classifier, we were able to generate a new electoral map shown in Fig. 3. For this map, a candidate wins a state if in that state there are more tweets favoring them as predicted by our classifier. We evaluated our classifier by the accuracy of the development set. We saw that our algorithm gave us a **89%** match with the election results for states which have more than 150 tweets. States with the number of tweets lower than 150 were dropped from our final output due to insufficient data.

Based on our new classifier, we decided to pick the most controversial tweets we saw during our analysis and check if we see any change in their category after our new features and model are used. As shown in Table 8, we were able to successfully shift the tweets from the incorrect category to the correct one.

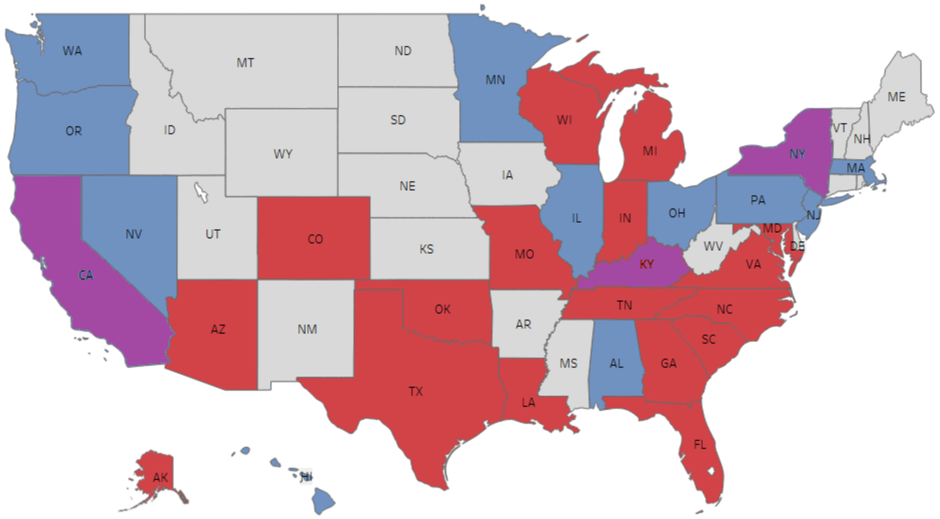

Figure 3: Electoral map based on our classifier on preprocessed tweets. Red states indicate Trump won based on our analysis and blue states indicates Clinton won. Grey states are states where we had less than 150 tweets combined across both candidates. The three purple states: CA, NY, KY, are used in the training set.

Table 8: Comparison between initial vader analysis and result from the classifier.

Tweet	Initial Vader Result	Classifier Result
Do you want a You're Hired president in Hillary Clinton or do you want a You're Fired president in Donald Trump	Trump	Hillary
It's smart to follow the laws: Clinton didn't follow the rules of saving her records as a paid public employee.	Hillary	Trump
I've got a question Tim... Should Hillary Clinton face consequences for mishandling classified information? #VPDebate	Hillary	Trump
Question is *grossly* misleading – @BudgetHawks said Clinton plan nearly pays for itself, Trump piles on debt!	Trump	Hillary
Trump managed to find the only politician that's dumber than him to be his running mate Impressive.	Trump	Hillary
Hillary Clinton's ”uninformed” #BasementDwellers know that Hillary has taken massive bribes from drugs companies to price gouge customers.	Hillary	Trump
Pence is insuring that Trump looks like a douche in the next debate.	Neutral	Hillary
No privatization of SS; I've paid in full, I want whats mine! Yes Clinton/Kaine; No Trump/Pence-not presidential!	Trump	Hillary

5.1 Clustering Tweets based on Topics

To gather an understanding of the topics that were being spoken about during the presidential elections we decided to analyse the tweets to find topic clusters based on the states where the tweets originated from. We wanted to see if we would be successful in finding patterns amongst states about subjects and topics that the tweets represent. Fig. 4 shows our final result for the state clustering.

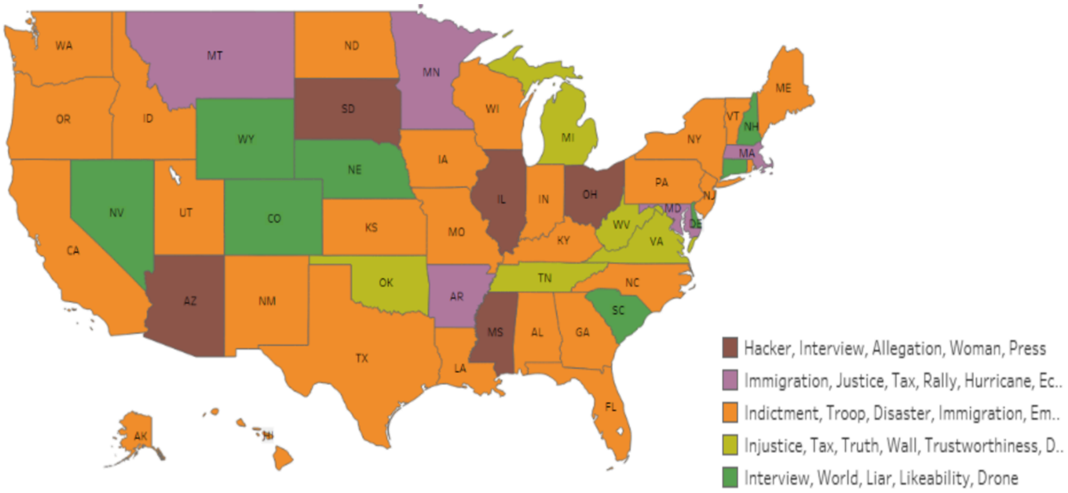
WA
MT
ND
MN
OR
ID
WY
SD
WI
MI
ME
VT
NH
NY
MA
NE
IA
IL
IN
OH
PA
NJ
NV
UT
CO
KS
MO
MD
DE
WV
VA
KY
CA
OK
AR
TN
NC
SC
AZ
NM
MS
AL
GA
TX
LA
FL
AK
HI
🟫 Hacker, Interview, Allegation, Woman, Press
🟪 Immigration, Justice, Tax, Rally, Hurricane, Ec..
🟧 Indictment, Troop, Disaster, Immigration, Em..
🟨 Injustice, Tax, Truth, Wall, Trustworthiness, D..
🟩 Interview, World, Liar, Likeability, Drone

Figure 4: Statewise clusters based on topics in tweets.

Some Other Observations and Learnings:

1. The Count Vectorizer (not shown in the code for the sake of brevity of document) does poorly in comparison to the TF-IDF Vectorizer.

2. We thought of using Word2Vec in place of the vectorizer that we used for our model, but a deeper analysis reflected that it wouldn't be a good practice to do so, since the word2vec model with dimensionality reduction would have resulted in clusters and using this as the input to our K-means clustering model would distort the results that we were expecting and probably give a fallacious holistic view.

6 Additional Consideration and Reflection

As part of our project, there were certain key things that we learnt throughout as part of our analysis. The first thing was that natural language processing is hard. There were multiple moments where we were challenged by the language to tweak and modify our algorithm to help account for edge cases and structures that arent as common in english written text but common in Tweets. It was very important that we spent time understanding the final result of our pre-processing and thus, spend time to clean our tweets and segregate important variables properly.

As mentioned above in our report, one of our major pivot points were when we realised that for analysis simple sentiment extraction of the tweets would not help us give relevant results in terms of who the tweet favors. As a result, we realised we need to spend time and utilize multiple NLP concepts to find sentiments towards certain subjects within the tweet.

Our pivot point:

COULD WE RECREATE THE ELECTORAL MAP BASED ON TWEET SENTIMENTS OF VOTERS?

TO

COULD WE BUILD A SUBJECT-BASED SENTIMENT ANALYZER TO IDENTIFY WHICH CANDIDATE A TWEET FAVORS?

We understood that we need to identify if a tweet is pro hillary or pro trump. For the purposes of the project we decided to scale down our categories to just Pro-Trump and Pro-Hillary, which we agree might not be the ideal case shown in Fig. 5.

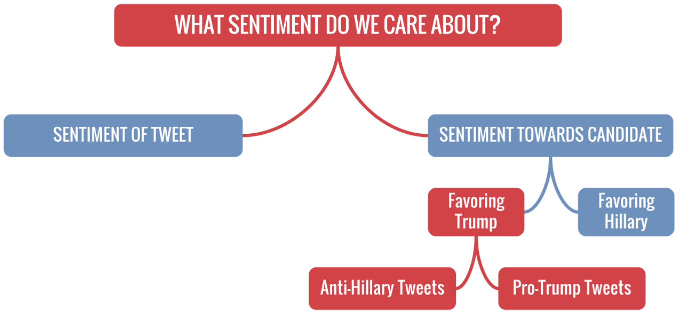

Figure 5: Breakdown of sentiment analysis and the possible actual categories.

During our project we saw that a generic subject/context based sentiment analysis approach would be very hard and might not prove fruitful for our case, and thus we decided to approach the problem by start to engineer our features based on the tweets to help establish a subject based classifier.

It is important to note that feature engineering needs to be carefully handled since we need to find natural language patterns that do not overfit on our train/test set. We looked at the structure of our tweets to understand the language being used to help dictate our features. There were points where we had to remove certain features from our model since we believed they might be skewing the results of our classifier. This was an important lesson and helped us understand the importance of extracting and selecting features.

Throughout our project, we had some really interesting findings which were somewhat similar to patterns that we have seen and heard about during the past couple of months. This served as an affirmation for what we were doing. We saw that:

- Twitter behavior during the 2016 elections has been predominantly negative and attacking.
    - It was very difficult to find tweets that were pro towards any particular camp. We saw much higher number of tweets that were mostly negative towards either of the candidates.
- A comprehensive majority of tweets favoring Trump are actually anti-Hillary, not pro-Trump.
    - Something that we saw during our analysis was that tweets were not favoring Trump but rather arguing against Hillary. Voters didnt seem to be particularly positive about a Trump presidency but showed dissatisfaction and negativity towards the possibility of a Hillary Presidency.
- To a fair extent, Twitter behavior has been consistent with the voting behavior of US states.
    - As we saw with our results we got an 89% match based on our twitter analysis and this sort of highlighted the similarity in the way people voted versus their sentiments expressed in the tweets.

7 Future Work

1. We need to collect more data to get higher number of geotagged tweets, which would help us achieve better training. We also see that tweets that are available are mostly anti hillary rather than pro trump and as a result that ends up skewing the model if not taken into account. We look at capturing more tweets which are pro trump.

2. We wish to apply this methodology to future elections, as well test on past elections by changing the subjects to see if the model stands up to the results achieved during this project.
3. We are still looking at discovering more features that can be used further help train our model. During our feature engineering we still believe that if we can gather higher number of tweets, features such as that hashtags can be used effectively to help get more features.
4. We aim to look at more complex algorithms within context based sentiment analysis. We believe that understanding and implementing those algorithms might push our results by creating a much more generic algorithm that helps analyse text based on the context.
5. We aim at normalizing the data by taking into account variables such as number of tweets by a single user, number of retweets, number of cross talk tweets between users, social influence on twitter, etc.
6. We aim to build a better classifier which helps distinguish within multiple categories which would include:
   - Pro-Clinton
   - Pro-Trump
   - Anti-Clinton
   - Anti-Trump
   - Neutral

8 Member Contribution

1. **Shirish Dhar**
   (a) Manually tagged some tweets.
   (b) Conducted pre-processing of tweet data including tagging, tokenizarion, extracting locations and hashtags from the dataset.
   (c) Computed and extracted NLP features from tweets to pass into the training model for our self-created sentiment analyzer.
   (d) Contributed to the building, parameter tuning and testing of the training models (primarily Neural networks) to ensure high accuracy on the test set.
   (e) Worked on setting up the online python web application for the project.
2. **Boris Lo**

(a) Manually tagged some tweets.

(b) Performed state-wise sentiment analysis of tweets using the Vader sentiment analyzer.

(c) Generated the topic feature.

(d) Aggregated the feature extraction codes together and pipelined it with the machine learning algorithm.

(e) Worked alongside others on the parts of speech analysis for the tweets to help create the bag of word feature based on usage of adjectives, verbs and nouns.

3. **Harman Shah Singh**

(a) Pre-processing of the Data.

(b) Manually tagged tweets. (amounting to over 1000+ tweets)

(c) Clustering tweets by state to come up with the top trending topics in each state and then clustering states together based on topics.

(d) Visualizing the results of clustering, sentiment analysis using Vader and our classifier using Tableau.

4. **Parv Sondhi**

(a) Worked on manually tagging subset of the tweets

(b) Worked on pre-processing twitter data to segregate hashtags as a separate usable entity for analysis

(c) Performed feature engineering for n-gram vicinity sentiment of the tweets and use that to pass data to the model

(d) Worked alongside others on the parts of speech analysis for the tweets to help create the bag of word feature based on usage of adjectives, verbs and nouns.

(e) Worked on setting up the flask/javascript web application for the users to interact with our project online.

In addition to the individual contributions, everyone also discussed and helped select relevant features, coming up with more ways to utilize NLP concepts, for the machine learning model and interpreted the results from our various analysis to help propel us forward as the project progressed.

References

[1] "2016 presidential election data hackathon." `https://github.com/WiMLDS/election-data-hackathon/tree/master/candidate-tweets-oct-2016/data`. Accessed: 2016-11-13.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[3] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python*. O'Reilly Media, Inc., 2009.

[4] O. Kummer, J. Savoy, and R. E. Argand, "Feature selection in sentiment analysis," 2012.

[5] M. Z. Asghar, A. Khan, S. Ahmad, and F. M. Kundi, "A review of feature extraction in sentiment analysis," *Journal of Basic and Applied Scientific Research*, vol. 4, no. 3, pp. 181–186, 2014.

[6] J. S. Manjaly, "Twitter based sentiment analysis for subject identification," *International Journal of Advanced Research in Computer and Communication Engineering*, 2013.