Perspective Analysis on U.S Presidential Candidates - 2016

Keshav Sridhar
School of Informatics & Computing
Indiana University, Bloomington
ksridhar@iu.edu

Brahmendra Sravan Kumar Patibandla
School of Informatics & Computing
Indiana University, Bloomington
bpatiban@iu.edu

Veera Marni
School of Informatics & Computing
Indiana University, Bloomington
vmarni@iu.edu

Yashwanth Konduri
School of Informatics & Computing
Indiana University, Bloomington
ykonduri@iu.edu

Qiwen Zhu
School of Informatics & Computing
Indiana University, Bloomington
qiwzhu@iu.edu

Abhimanyu Reddy Maddireddy
School of Informatics & Computing
Indiana University, Bloomington
amaddire@iu.edu

Venkata Pradeep Katrevula
School of Informatics & Computing
Indiana University, Bloomington
vkatrevu@indiana.edu

Abstract

US presidential elections are always of interest across the globe. Since 2012, a significant portion of the election campaigning is being undertaken on social media where candidates rally for support and take occasional swipes at rival candidates. People across the globe also weigh in their opinions. In this paper we show that by gathering sufficient amount of data from social media like twitter we can highlight the direction of the campaign of the Presidential candidates and further analyze the user's perspective on the current election campaign. We have also visualized our work using word clouds in the form of candidate's faces and also included a US map on which we show state wise interpretations.

1 Introduction

We all know that the election campaigns are hotly contested affairs between the candidates lobbying to be their respective party's presidential nominee. In general, the race for white house begins well ahead of time with candidates from different parties projecting their opinions to grab support from the public. Most of the candidates use social networking to convey their stand on

various issues. This project of ours is to find out how each candidate is projecting their stand and the issues that they consider to be most important according to them. Twitter becomes a major source of data to analyze the direction in which a candidate campaign is moving as most of the candidates keep their followers updated in the social media space. The general statistics will not convey enough information to the reader until it has been visualized. So, as a part of our project we considered visualization as an important aspect and built visualizations that are browser compatible using D3.js.

The rest of the paper is organized as follows: First, we briefly discuss about the related works that inspired us. Second, we explain the data collection, storage and retrieval operations used by us for the development of this project. Third, we will discuss the various methods we have used in the course of this project and the results obtained from them. Fourth, we discuss the visualization techniques used by us to highlight our results to the end-user. And finally we will close with the conclusion and the future work that can be implemented to further improve upon what we achieved.

2 Related Work

This project is inspired from the work of IU students Ritesh Agarwal et.al. , "Analyzing US Presidential Elections of 2016". We have also looked into some other works to learn about standard visualization standards. The algorithm to customize the shape of the word cloud, developed by Timothy Guan−tin Chien is used in this paper. The source for this algorithm can be found in the book *Beautiful Visualization* [4]. Also, the high level implementation of this algorithm is discussed later in this paper.

"Predicting US Primary Elections with Twitter" [1] illustrates the architecture for real time processing of twitter streams. Another work reviewed was, "Sentiment Analysis of Political Tweets: Towards an Accurate Classifier" [2] which explains about how to deal with political tweets with using supervised machine learning techniques. "Predicting US Primary Elections with Twitter" [1] highlights how to deal with various features of a tweet and how they can be used. This paper highlights the uses of using a NoSQL database which helps retain information easily without loss. This prompted us to use MongoDB instead of traditional Relational databases.

3 Data Collection

In the paper, we primarily deal with the following aspects of the election: (1) identifying the ideology of a given tweet, (2) identification of public reaction, (3) identification of campaign focus of candidates. Each goal requires a special set of data for building models. The source(s) and method(s) of extraction of the data for each goal heavily influence the result.

The first goal can be supported from data obtained as tweets from Twitter. All available tweets were extracted from the timeline of each of the Presidential candidates, namely, Hillary Clinton (@HillaryClinton), Bernie Sanders (@SenSanders), Donald Trump (@realDonaldTrump), Ted Cruz (@tedcruz) and John Kasich (@JohnKasich). For extracting tweets from user timeline, we used Timeline function provided by Twitter's REST API.

The second goal can be achieved by analyzing public reaction during the campaign. To obtain data on public reactions, we turn to Twitter once again. For public response, we extract tweets based on the hashtags associated to the candidates and US Elections in general. A hashtag is a special word/phrase preceded by a pound symbol, used in social media websites. Hashtags help in identifying messages on a specific topic. The hashtags used to extract data were selected in a careful manner. A set of general hashtags like #USElections, #UsElections2016, #DemCaucus, #GOP were identified and approximately 6000 tweets were extracted. For extraction of tweets with these hashtags, we used the Search function provided by the API.

From the 6000 tweets extracted above, we identified the frequently used hashtags. These hashtags could be related to Primary of a state at that given time, particular to a candidate (#TrumpTrain etc.). Using those hashtags we further extracted tweets using Search function of API. After all the extractions, we have a corpus of ~130,000 public tweets.

The official twitter REST API provides access to their data from their portal. But, the API has certain restrictions on the access of data. The restrictions are: (1) when using Timeline function, only a maximum of 3200 recent tweets can be accessed from the timeline of the target twitter handle, (2) when using Search function, tweets from the past 10 days can only be extracted, (3) a

maximum of 15 requests can be made in a 15 minute window. Such limitations on the API make tweet extraction a very slow process.

We used Python to program tweet extraction. TweePy is a module available for Python, which makes use of the REST API provided by Twitter. The data returned by the API will be in JSON format. Usually, many papers which deal with tweets, save the text, username, and date in a spreadsheet and discard all other features. However, we didn't want to lose any data since it is very difficult to obtain the discarded data in future, if needed. Hence, we needed to find a way to save this data in its pristine form. For this, we had to find a database which is schema-less and document based. Schema-less nature is required because the structure of JSON could change based on the structure of tweet (ex. Username mentions etc.) MongoDB proved to be a useful database which comes with schema-less and document based features.

To connect to MongoDB server using Python, we used PyMongo module. MongoDB has the following structure. JSON documents are stored in collections, and one or more collections form a database. For this paper, we created 6 collections. Each Candidate has a collection for his/her timeline. The last collection (tweetStream) is used to store tweets pertaining to public.

The collections are named as follows:

Description of Collection	Collection Name
Bernie Sanders	SenSanders
Hillary Clinton	HillaryClinton
Donald Trump	realDonaldTrump
Ted Cruz	tedcruz
John Kasich	JohnKasich
Public Response	tweetStream

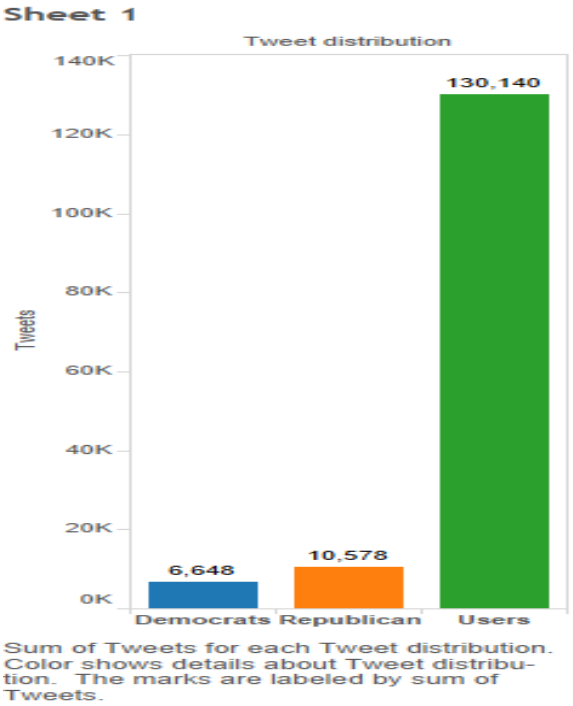


Figure 3.1 Total number of tweets captured for each distribution

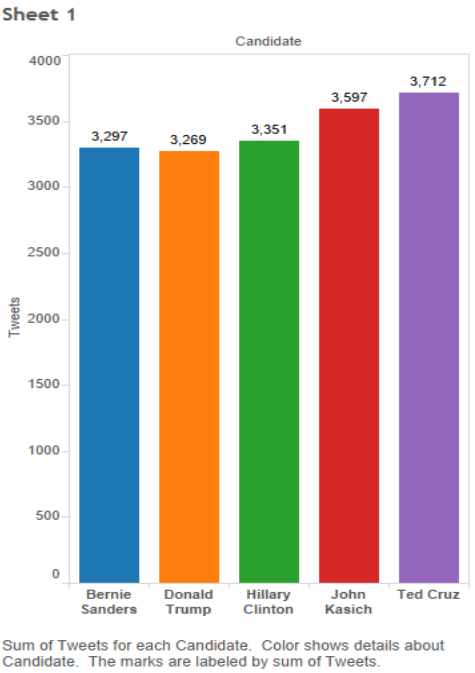


Figure 3.2 Data distribution of the candidate tweets captured

To achieve the 3rd goal of identifying candidate campaign focus, we must collect a corpus of data which comprises of the speeches, press releases tweets from the campaign offices of each

Presidential Candidate. For this, we scraped data from the Election Repository hosted by University of California, Santa Barbara[5], candidate campaign websites. This data is stored as plain text files.

So far, all the data estimated to be essential has been obtained.

4 Methods

Instead of trying to predict who will win the election, we wanted to predict whether a tweet was tweeted by a republican candidate or a democratic candidate and perform further fine grain analysis on which candidate would the tweet belong to. We split the data as 70% train and 30% test data.

We came up with two main tasks or goals while preparing this data, namely,

- Classification task
- Modelling task

4.1 Classification task - SVM

Our first task was to build a classifier to predict the "perspective" of a given tweet i.e., to assign a suitable class (Republican/Democratic or Username). We use a supervised learning machine learning technique (SVM). We try to develop two classifiers. Classifier 1 predicts the Party/ideology which is close to the given tweet. Hence the classes are Democratic or Republican. Classifier 2 predicts which user is more likely to tweet such tweet. This classifier has classes which are the names of the Presidential Candidates (5 classes).

For the SVM model that we build, we input word vectors built from a bag of words using the words from the training dataset. First we consider only the tweets tweeted in English language, then we clean our data by using regular expressions to snip the website URLs and other webpage links (this is useless data for our task of predicting the perspective of the tweet), we then remove the user twitter handles, remove stop words from each tweet by utilizing stop words from NLTK package and supplementing them with additional words.

We build word vectors using NumPy arrays and bag of words data and passing them to our classifier. We represent the class of republican candidates as '0' and the class of democratic candidates as '1'. We achieve the results as shown in Figure 4.1.

```
Accuracy score:
0.83820398684
Precision score:
0.783823529412
Recall score:
0.801905717151
Classification report:
             precision    recall  f1-score   support

          0       0.87      0.86      0.87      3173
          1       0.78      0.80      0.79      1994

avg / total       0.84      0.84      0.84      5167
```

Figure 4.1 Republican-Democratic SVM Classifier Results

We then proceed to build a fine-grained classifier for the same task, but instead this time we want to predict a tweet corresponding to each candidate. We use OneVsRest SVM classifier to build our model and then predict. For classification purposes, we assign 1 – Hillary Clinton, 2 –Bernie Sanders, 3 – Donald Trump, 4 – John Kasich, 5 – Ted Cruz. We achieve the results as shown in Figure 4.2

```
Accuracy score:
0.745936532508
Precision score:
0.748824472608
Recall score:
0.745936532508
Classification report:
             precision    recall  f1-score   support

          1       0.72      0.73      0.73      1005
          2       0.84      0.79      0.82       989
          3       0.78      0.70      0.74       981
          4       0.71      0.75      0.73      1079
          5       0.70      0.75      0.73      1114

avg / total       0.75      0.75      0.75      5168
```

Figure 4.2 Fine grained classification results

The reason we get a good model apart from the data cleaning is because of the even distribution of the train data for our classifier during the data split procedure. The way we split the data before the processing step is by ensuring that we take 70% of data from each class and merge them to form our training data. In this way, we get equal amount of training data for each class. Also, we believe each candidate's twitter data is proliferated with hashtags pertaining to their campaign mantra, which makes the data cluster easily.

4.2 Modelling task – LDA

For our topic modelling task, we chose to utilize the Latent Dirichlet Allocation model

implementation provided by genism[3]. In this part, we don't process the text as much as the previous section. We use the data collected from the candidate's speeches and the major chunk of data from the public tweets to assess the topics related to each candidate. We find some interesting details about each candidate once we use topic modelling on the combined data of the candidate speeches along with the public tweets extracted in relation to each candidate.

```
'0.115*demdebate + 0.037*bernie + 0.030*feelthebern
 + 0.027*vote + ''0.027*newyorkprimary + 0.024*imwithher
 + 0.020*hillary + ''0.018*berniesanders + 0.017*newyork
 + 0.014*take'
```

Figure 4.3 Here we can see both democratic candidates fighting for the New York primary (which was won by Hillary Clinton incidentally)

```
'0.058*primaryday + 0.030*htt + 0.028*trump
 + 0.026*going + 0.016*believe + '"0.015*i'm
 + 0.012*primary + 0.010*you'r
 + 0.009*nevercruz + 0.008*muslim"
```

Figure 4.4 We can see here in one of the topics, the words "trump","nevercruz" and "muslim" are closely related topics which were mined from public tweets.

```
 '0.044*amp + 0.033*vote + 0.031*cruz
  + 0.030*voted + 0.027*polls + 0.027*ted '
" + 0.022*let's + 0.022*cruzcrew
  + 0.021*election + 0.019*thank"
```

Figure 4.5 Ted Cruz's Crew!

```
'0.022*making + 0.020*war + 0.019*much
 + 0.015*iraq + 0.014*dem + '
'0.014*muslim + 0.014*integrity +
 0.014*issues + 0.013*trump + ' '0.012*endorses'
```

Fig 4.6 Another topic modelled on public tweets

5 Visualization

We visualized the results of these classifications and the topic modelling using D3.js. There are two major categories of visualizations. The first bound results of analysis for each candidates' tweets with each of five candidates. These visualizations are presented in the form of candidates' line-drawing portraits. We first generated path data by exploiting an algorithm that can customize the shape of word cloud, proposed by Timothy Guan-tin Chien[6]. As Jonathan Feinberg invented the algorithm in his book Beautiful Visualization: Chapter 3, the basic idea behind this is: 1) place word randomly in somewhere on the path, 2) if the word intersects with any previously placed word, move it one step along an every-increasing spiral, and 3) this process is repeated until the path is fully filled. As a result, the path of these line drawings are filled with results of LDA topic modelling analysis for the five candidates' tweets, speeches respectively, as shown in Figure 5.1.

Figure 5.1: Line-drawings of two candidates, with dark color part filled with each analysis of their tweets respectively.

The other visualization is in the form of U.S. map (exclusive of Alaska and Hawaii). This visualization shows the analysis results from the tweet Stream data, the users' tweets, as shown in Figure 5.2. Because the map itself is filled with dark color for the territory, thus data from the analysis results made up the map after visualizing.

The map based visualization shows that the most talked-about candidate among the five is unsurprisingly Donald Trump, with Ted Cruz following in second largest font. This may indicate the pro-Republic trend, at least, in the cyberspace. This is in accordance with the classification results that more than 60% of user tweets are about Republican Party.

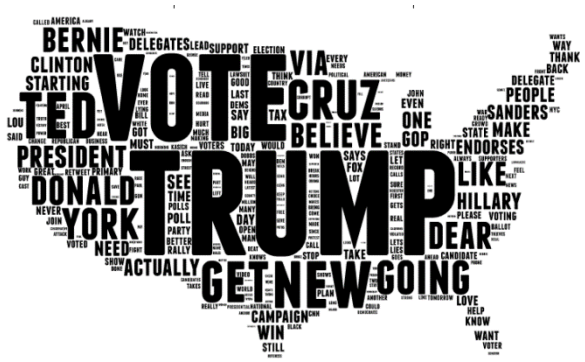


Figure 5.2

6 Conclusion and Future Work

Based on the results documented, we believe that pre-processing the data, cleaning the data and having an even distribution of data for training in case of multiple classes, has a major impact on the efficacy of the model.

Twitter data, speech data of the candidates can be used effectively for topic modelling.

- We can build a website that houses all the analytics and visualizations.
- Utilize a streaming service to predict the perspective of a tweet in real time.
- Build a neural network that "learns" more about each candidate.
- Utilize unsupervised learning techniques.
- Utilizing higher level n-grams to get a better grasp of the topics for each candidate.
- Use geo-location data to aid in twitter analysis of the public tweets.

References

[1] Predicting US Primary Elections with Twitter. http://snap.stanford.edu/social2012/papers/shi.pdf

[2] Sentiment Analysis of Political Tweets: Towards an Accurate Classifier.

http://www.aclweb.org/anthology/W13-1106

[3] Software Framework for Topic Modelling with Large Corpora - Radim Reh˘u˚rek and Petr Sojka

[4] Beautiful Visualization by Jonathan Feinberg

[5] 2016 Presidential Elections Documents http://www.presidency.ucsb.edu/2016_election.php

[6] http://timdream.org/wordcloud2.js/

Group Members

Qiwen Zhu, Master of Information Science (2015-2016), SOIC, Indiana University. I am pursuing my Master of Information Science with a concentration on Data Science in School of Computing and Informatics. I have specialized in Database management systems and their applications. I spent 8 years working as a system administrator in a government agency of radio spectrum management in China, managing a team of database maintenance and office network administration. Huge amount of radio spectrum in my previous work intrigued me to explore in the field of data science. My first degree was in Information System and Management from Lanzhou University of Finance and Economics.

Qiwen's linkedin

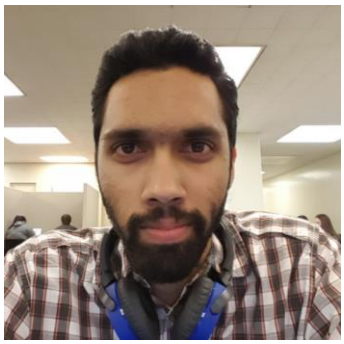
Veera Marni, Master of Data science at Indiana University Bloomington (2016-17). I have previously worked on Database using SQL and PL-SQL for a year. I am interested in Image analysis and machine learning. I also like to work on projects from Kaggle. Most of my works can be found on GIT-hub https://github.com/narayana1043

Abhimanyu Reddy, Master of Data Science at Indiana University Bloomington(2016-17). I worked on prediction modeling of survey data, trend and sentiment analysis of social media data for a major technology company. I am proficient in using R, SAS, and Python. I am interested in predictive analysis of data, including social data. I participate in Kaggle competition and hackathons.

Venkata Pradeep Katrevula, Master of Computational Science student at Indiana University (2016 -17). I have two years of development experience in web applications. I have worked on developing Medical portal applications and factory applications.I am interested in Machine learning and application development.
This is my linkedin profile link https://www.linkedin.com/in/venkata-pradeep-k-82010141

Yashwanth Konduri, Master of Data Science at Indiana University, Bloomington (2016-17). I have worked as a Business analyst at Deloitte Consulting. I worked previously on gathering requirements, creating SAP Business object reports and worked on Informatica. I currently am a student of data science and interested in Machine learning and data mining. I love to gain previously unknown useful insights from data and present them in a clear and appealing manner. My Linkedin profile: https://www.linkedin.com/in/yashwanth-konduri-226b65bb

Brahmendra Sravan Kumar Patibandla, Master of Data Science at Indiana University, Bloomington (2016-17). I worked as a Database engineer in an Analytics startup. I did my graduation from ISM Dhanbad, India with Electronics as major. I am interested in Data Management, Machine Learning and IOT. My LinkedIn Profile.



Keshav Sridhar, M.S in Data Science, Indiana University Bloomington (2016-2017). I have worked as a Business Technology Analyst for 2 years at Deloitte. My interests include playing and watching soccer, playing video games (RPGs,MMOs,RTSs). On a professional end point I love to find patterns in data and try to gather some insights from them.

LinkedIn:https://in.linkedin.com/in/keshav-sridhar-b0088953