FEAT:
A FACEBOOK EXTRACTION AND ANALYSIS TOOLKIT

A Thesis
by
HAIHOUA YANG

Submitted to the Graduate School
at Appalachian State University
in partial fulfillment of the requirements for the degree of
MASTER OF COMPUTER SCIENCE

December 2016
Department of Computer Science

FEAT:
A FACEBOOK EXTRACTION AND ANALYSIS TOOLKIT

A Thesis
by
Haihoua Yang
December 2016

APPROVED BY:

Cindy Norris
Chairperson, Thesis Committee

R. Mitchell Parry
Member, Thesis Committee

Rahman Tashakkori
Member, Thesis Committee

James T. Wilkes
Chairperson, Department of Computer Science

Max C. Poole, Ph.D.
Dean, Cratis D. Williams School of Graduate Studies

Abstract

FEAT:
A FACEBOOK EXTRACTION AND ANALYSIS TOOLKIT

Haihoua Yang
B.S., Appalachian State University
M.S., Appalachian State University

Chairperson: Cindy Norris

Social media usage has become mainstream. According to a recent study done by Edison Research in 2016, 78% of the U.S. population has a social media profile [8]. The number of active Facebook users is over one billion. In addition, 71% of adults use Facebook, which is the target of this thesis. Because Facebook is so widely used, it is also a popular medium for those wanting to promote their products and ideas, including presidential candidates. Many researchers have extracted data from social media sites, including Facebook, to predict the outcome of elections, to predict election turnout by political party, and to determine voter opinions. This thesis will discuss the development and use of a suite of tools for gathering and analyzing data collected from the social media site, Facebook. Although the suite of tools can be used to collect data from any public Facebook site, this thesis will specifically focus on using the tools to extract data from the pages of presidential candidates. In addition to extracting Facebook data and storing the data in a database, tools in the suite can be used to analyze and visualize the collected data.

Acknowledgments

First, I would like to thank my advisor, Dr. Cindy Norris, for the guidance and insight she provided. Dr. Norris provided patient assistance in the research and writing of this thesis. She supplied multiple resources and motivation throughout the process. I cannot imagine completing this thesis without her continuous support. Second, I would like to thank my committee members, Dr. Parry and Dr. Tashakkori, for their time and effort. They have both been both resourceful and patiently assisted me in the completion of this thesis.

Third, I would like to thank Timothy Johnson for helping me in designing my website. He has been very enterprising in working with the Django framework and Chart.js. He spent a significant amount of time working with me to design graphs to display the results and create forms to configure what graphs appear on the website. Without him, the design and implementation of the website would not have gone smoothly as it did.

I would also like to thank my peers and family for supporting me throughout the years and encouraging me to push myself to reach new limits. Last, I am very grateful to the NSF S-STEM Program that provided me with funding throughout my undergraduate and graduate studies at ASU.

Table of Contents

Chapter 1 - Introduction

Social media sites are now widely used by politicians, companies, private organizations, and individuals to advertise events, products, and ideas. In the 2008 election, President Obama demonstrated the power of social media and how it could be used to "change the political landscape" [5]. Particularly, during that election social media platforms became the leading mechanism for engaging voters of younger ages and reaching minorities. The powerful use of social media allowed Obama an overall advantage over McCain on attracting voters and rallying turnout. In contrast, McCain's inefficient management of social media resulted in small outreach. Facebook continues to be a popular medium for political candidates to spread their influence. For example, the Maryland Presidential Primary Election State Candidate List of 2016, lists 22 candidates. Of the 22 candidates, 17 have listed Facebook pages [15].

This thesis discusses the extraction and visualization of data from the Facebook pages of presidential candidates. We describe a Facebook Extraction and Analysis Toolkit (FEAT) that uses the Facebook API to gather data and provides tools to analyze and visualize the collected data. Listed below are the candidates and the Facebook URLs that are used in this thesis:

- Hillary Clinton - https://www.facebook.com/hillaryclinton/
- Donald Trump - https://www.facebook.com/DonaldTrump/

The data extracted was obtained for about a period of ten months from January 2016 to November 2016. The following tools are included in the project:

- FEAT Extractor that collects data from Facebook and stores the data into a database

- FEAT Analysis Tools that
    - count word frequencies
    - calculate Readability scores
    - determine the sentiment of comments and posts
- FEAT Visualizer that displays graphs of the analyses that were performed

The data collection tools use the Facebook API to gather data such as the numbers of followers, posts, comments, and shares from Facebook pages and then store the data into a MySQL database for later analysis.

Many systems today provide an Application Program Interface (API) that specifies how new software can request services from an existing system. APIs abstract the underlying implementation and only expose entities that the developers may need, making it easier for developers to use the program to create applications. APIs describe the fields and functions that allow the developer to interact with the underlying software. For example, the Graph API provided by Facebook allows developers to obtain, insert, and delete information on Facebook. Some of these operations are restricted to developers with specific permissions.

The data obtained via the Facebook API is stored into a MySQL database. MySQL is an open-source Relational Database Management System (RDBMS). Data in a relational database is organized into tables; each of the tables has a well-defined relationship with the other tables. MySQL software interacts with the user to perform insertion, deletion, and modification of data in the database.

Visualization tools extract the data from the database and create graphs using Chart.js. Chart.js is a JavaScript charting library that uses the HTML5 canvas to render the graphs. There are numerous charting libraries available such as D3.js, FusionCharts, and

Highcharts, however Chart.js is perfect for supplying simple and elegant charts. In addition, Chart.js is open source and is easy to use.

The remainder of this thesis is organized as follows. Chapter 2 examines other works that are related to the use and analysis of social media. Chapter 3 provides background information about text analysis. Chapter 4 discusses the methodology that was used to extract data from Facebook and the tools use to analyze the data. In addition, Chapter 4 also discusses how the website was implemented and used. Chapter 5 presents the analysis of the data obtained from the presidential candidates' Facebook pages. Chapter 6 provides the summary and conclusion for the research.

Chapter 2 – Related Work

Numerous social media organizations, such as Facebook, Twitter, Instagram, LinkedIn, and Pinterest, provide an API that allows a software developer access to some of the social media data. For instance, the Facebook API is structured like a graph with nodes and edges. A node represents content on the page such as a post or image. An edge provides a link between related nodes such as a post and a comment on that post. However, an API may not provide all the data a developer may wish to obtain. For example, the Facebook API allows the extraction of a comment but not the commenter information such as gender and age that may be public information on the commenter's page. The Facebook API will not allow access to personal pages. Also, an API may place a restriction on how many consecutive queries are allowed per day. In these cases, web scraping can be used to crawl through web pages to look for desired information. Since a web scrapper does not restrict access in the way an API does, a developer will have more freedom in obtaining data from web pages.

Web scraping is a technique employed to automatically extract data that can be viewed on a website. Web scraping is similar to web indexing in that it uses bots or web crawlers to traverse websites looking for specific HTML tags or specific labels. For example, Facebook displays a birthdate after the label Birthday that can be obtained by scraping the page, searching specifically for that label. The structure of HTML makes it possible to retrieve and organize data that is viewable on a web page. Web scraping is commonly used in online price comparison, weather data monitoring, website change detection, and social media research.

This chapter discusses tools that use APIs and/or web scraping to obtain social media data. In addition, the chapter describes the results obtained by analyzing the data. These works are divided into two sections. Section 2.1 describes tools and research unrelated to the use of social media for political gain and insight. Section 2.2 describes research into the use of social media for political purposes.

2.1 Analysis of Social Media Data

According to Edison Research, in 2016, 78% of the U.S. population has a social media profile [8]. The number of active Facebook users is over one billion. Instagram has over 400 million users, and Twitter has over 320 million. For these reasons, social media has become popular for business and political outreach. In addition, organizations are using social media in increasingly more sophisticated ways. For example, businesses can provide very targeted advertising that is based upon social media use, and emergency management organizations can use social media to track crises and render aid. This section describes some recent uses and analysis of social media.

The TweetTracker project uses the Twitter Stream API to assist Humanitarian Aid and Disaster Relief respondents [13]. Twitter data is stored in a database that is processed to pinpoint the location of tweets. The software produces a map on a webpage; markers on the map identify the locations from which the tweets originated. In addition, an application runs in the background that monitors the Twitter streaming feed for specific keywords and hashtags. The TweetTracker tool was used during the Haiti cholera outbreak. The tweets were filtered for “#cholera” and visualized on a map with colored tags. Green tags were used for geo-located tweets (tweets that contain location information) and blue tags were set for

non-geo-located tweets. The TweetTracker tool can provide emergency responders with site information immediately after a crisis.

He et al. performed text mining to analyze the use of social media by three large pizza chains: Domino's, Papa John's, and Pizza Hut [10]. Text mining is a technique to create meaningful information from unstructured text, in this case, Twitter and Facebook. The process to obtain the results was separated into three steps. The first step was to extract and prepare the data from Facebook and Twitter. The data compiled included the number of fans/followers and post/tweets in the month of October 2012. The next step performed the actual text mining of the data collected. Since there was no prior knowledge of categories, all tweets were combined to create one set and all posts were combined to create a second. Common themes were uncovered in each set. After the themes were defined, the posts and tweets were reexamined to separate them into groups based on the themes. Lastly, the researchers analyzed the data to determine how pizza companies could make better use of social media. For example, one recommendation was to constantly monitor their own and their competitors' social media. Monitoring social media helps the company judge how well its product is received by their customers and leads to a greater knowledge of the competitors' products. He suggested the use of free and commercial Internet tools to monitor certain websites. Some examples listed were Google Alerts, Social Mention, Quora, and HootSuite. These tools analyze webpages and provide graphical reports based on keywords, hashtags, sentiment, and influencers.

Asur et al. demonstrated how to use chatter from Twitter to forecast box-office revenues for movies [1]. The researchers used keywords associated with particular movies to search for related tweets. In total, they extracted 2.89 million tweets referring to 24 movies.

The data was then used to calculate the tweet-rate of each movie, which is the number of tweets divided by the number of hours the tweets were collected:

$$TweetRate = \frac{|Tweets\ of\ Movie|}{|Hours|} \quad 2.1$$

The authors used the tweet-rate to predict first-weekend box office revenues and found a strong positive correlation (with a correlation coefficient of 0.90) between average tweet rate and box-office gross. The predictions were evaluated using box-office information extracted from the Box Office Mojo website. For example, the movie *Transylmania*, had a low tweet-rate and was recorded to have the lowest-grossing opening for a movie playing at over 1000 sites. In comparison, the movie *Twilight: New Moon* made 142 million dollars and averaged 1365.8 tweets per hour.

Sentiment Analysis is commonly used to measure the attitude of reviews such as those of products and movies. Sentiment analysis on movie reviews can be readily performed given the extensive collections of reviews such as IMDb that can be found on the Internet. Also, reviewers often summarize their reviews by machine-extractable indicators such as the number of stars [19]. Asur and Huberman also performed sentiment analysis on their set of Twitter data to predict if a movie will be successful [1]. The ratio of positive tweets over negative tweets was used to improve their prediction strategy:

$$PNratio = \frac{|Tweets\ with\ Positive\ Sentiment|}{|Tweets\ with\ Negative\ Sentiment|} \quad 2.2$$

They predicted that if a movie has many more positive tweets than negative tweets then the movie to be popular. The PNratio and the tweet-rate were used together to predict box-office revenue for the second weekend. They found that the use of the PNratio did not improve the prediction. However, the tweet-rate alone was just as effective predicting box office revenues for the second weekend as it was for the first.

Pak and Paroubek investigated the performance of sentiment analysis on micro-blogs, specifically, tweets from Twitter [18]. The researchers used the Twitter API to collect 300,000 posts evenly split into three datasets: texts containing positive emotions, texts containing negative emotions, and texts expressing a fact. Texts categorized as positive contained emoticons such as **:)** or **:-)**. Texts categorized as negative contained emoticons such as **:-(** or **=(**. Texts expressing a fact were retrieved from the Twitter accounts of newspapers and magazines such as the New York Times or the Washington Post. After categorizing the tweets, the researchers performed a linguistic analysis of the datasets. The analysis found that objective texts tend to contain more common and proper nouns, and the subjective texts contain personal pronouns and simple past tense. In addition, positive texts more frequently use superlative adverbs such as "most" and "best" while negative texts tend to use past tense verbs such as "missed," "bored," and "gone." Next, the researchers extracted features from the datasets to train a sentiment classifier. In particular, the researchers trained the classifier using n-grams that were obtained from the datasets. These n-grams were formed by filtering each Twitter post to eliminate Twitter usernames, URLs, and emoticons; tokenizing the remaining post; removing stop words such as "a," "an," and "the"; and finally, concatenating consecutive words to create n-grams. In experiments with several classifiers, they found that the Naïve Bayes classifiers yielded the best results. The Naïve Bayes classifier that used n-grams and part-of-speech (POS) tags as features could classify positive, negative, and neutral tweets with a high accuracy.

2.2 Politics and Social Media

Social media has provided opportunities for politicians and social science researchers. Politicians are interested in using social media to connect with and influence voters.

Researchers are interested in determining the impact of social media on political opinion, and evaluating how users receive and share election information. This section discusses some of the research into these issues.

In the 2004, 2008, and 2012 elections, Facebook allowed users to tell their friends who they are voting for by writing a message that appeared at the top of their news feeds [2]. After the 2012 election, the Facebook Data Science team used the voting data entered by over 9 million users along with information about users to explore the relationships between voting and user attributes such as gender and party affiliation. The Facebook data indicated that women were far more likely to share who they voted for. This agrees with Facebook findings that, in general, women are far more likely than men to share information on Facebook. In addition, the researchers found that users who viewed themselves as liberals or democrats, were more likely to indicate who they voted for on Facebook.

Research performed by the Pew Research Center compared candidates' use of social media in the election of 2016 and the previous elections [21]. In one study, the research determined which social media sites were used by the candidates. In 2008, both Obama and McCain used Facebook, YouTube, Flickr, and Myspace. In 2012, Obama used ten social media sites while Romney only used three. In the 2016 election, Bernie Sanders, Hillary Clinton, and Donald Trump used at least four social media sites. In addition, the researchers collected and analyzed posts and tweets from Trump, Clinton, and Sanders Facebook pages and Twitter profiles from May 11 to May 31, 2016. The researchers found that all three candidates post at a similar rate but Trump receives the most response in terms of comments, shares, reactions, and retweets. Research also showed that Trump and Clinton focused on

each other on Facebook. Trump refers to Clinton in 38 posts and Clinton refers to Trump in 45 of her posts.

Scientists from Queen Mary University of London analyzed tweets by 10,000 users on Twitter [23,25]. The researchers separated users into political parties based on which American politicians' Twitter feeds the user followed. This allowed tweets made by the users to be categorized by political party. Two sets of tweets were analyzed for word frequency. Words that they found to be affiliated with the Democratic party included words such as "I," "me," "mine," curse words, and words that express positive sentiments. Republicans more frequently use the words "we," "our," "us," "God," and words that express negative sentiment.

Choy et al. developed two models using Twitter data to predict the outcome of the 2012 Presidential election [5]. The researchers collected 7,541,470 tweets between August 12, 2012 and October 31, 2012. The AFINN corpus was used to extract the sentiments from the tweets. Socio-demographic data and census information were used to correct for bias. The first model assumed that the Twitter information reflects the opinion of anyone active on the Internet. The second model assumes that the Twitter influence is limited to the Twitter population and that the rest of the electorate are better modeled by their prior political affiliation. The first model predicted a very tight race. The second model included party affiliation and predicted a comfortable win for Obama. Both models validated the possibility of using Twitter to predict elections.

In April, 2016, Conlen and Fischer-Baum created a website that displayed a map of followers of presidential candidates per location [6]. If the number of likes actually translated into votes, Bernie Sanders would have beaten Hillary Clinton by a nearly 3 to 1 margin.

According to Pew Research Center, 58% of American adults use Facebook [21]. However, Facebook users are not representative of the country, as most users tend to be young, low-income, and female. In addition, not all users that vote follow a candidate on Facebook. Although likes do not translate into votes, the map does indicate where candidates are likely to receive support. As can be expected, Sanders had large pockets of supporters in the northeastern states and Ted Cruz had pockets of supporters in Texas.

Unlike the research discussed in this chapter that targets specific politicians, elections, or analyses, this thesis discusses the design and implementation of a toolkit that can be used to extract and analyze data from any public Facebook page. Since 2016 is an election year, the tools are applied to the Facebook pages of presidential candidates, but these tools could be used to extract and analyze data from the pages of businesses, organizations or celebrities. The next chapter provides the background material needed to understand the analyses that are performed by the FEAT toolkit.

Chapter 3 – Text Analysis

Text analysis, or mining, techniques obtain information from text by looking for certain items such as the frequency of use of certain types of words. These techniques begin with processes to parse the text to extract important pieces of the text and discard others. This parsed text is then usually stored in a database. Next, the text is analyzed to detect certain patterns and trends. Finally, the output is evaluated. Text analysis techniques range from simple tasks such as counting the frequency of words to the more difficult tasks of determining the writer's attitude toward a particular topic.

Section 3.1 provides information about a type of text analysis known as sentiment analysis. Sentiment analysis is the process of determining whether a piece of writing is positive, negative, or neutral, thereby offering a method to determine a writer's opinion on a specific topic. For example, sentiment analysis can provide an automated approach for categorizing reviews of movies. Techniques for performing sentiment analysis are discussed in Section 3.1. Particularly, the section focuses on the Naïve Bayes model used in the FEAT toolkit.

Section 3.2 and 3.3 provides background information on word frequency, word cloud, and readability analysis. Word frequency graphs display the N significant words that appear most frequently in a document or set of documents. This analysis can provide an indication about what topics a writer or group of writers deem most important. A word cloud is a graphical technique for illustrating the frequency of the words. Readability analysis determines the ease at which text can be read and understood. It is widely used to choose

books for school age readers, but also can be used to provide insight into the literacy level of the writer.

3.1 Sentiment Analysis

Sentiment Analysis is the computational study of people's attitudes toward a topic by uncovering and categorizing a sentiment found in text. Specifically, sentiment analysis finds opinions in text, extracts the words or phrases that indicate an opinion, and classifies the text per sentiment expressed. The sentiment of each text is classified into categories of positive, negative, or neutral.

Sentiment analysis has become important to businesses as customers increasingly use the Internet to express their thoughts about different products [10]. Businesses can analyze product reviews on sites such as Amazon and make business decisions based upon these reviews. Sentiment analysis could also be applied to stock markets, news articles, and political debates. Social media is considered a good source of data for sentiment analysis because its environment allows people to freely share and discuss their opinions on a variety of topics. For example, micro-blogs such as tweets on Twitter, can be analyzed to extract opinions about certain candidates or political topics. This analysis can then be used to build a prediction for election results.

Sentiment analysis applications can be categorized into three levels: document-level, sentence-level, and aspect-level. Document-level sentiment analysis classifies the sentiment of an opinion document such as a movie or book review. Sentence-level sentiment analysis classifies the sentiment of the text by sentence. Fundamentally, techniques that perform document level sentiment analysis and sentence level sentiment analysis are the same since sentences are simply short documents. Aspect level analysis identifies both the sentiment and

the entity to which the sentiment applies. For example, aspect level analysis on the sentence "Michael Fassbender is handsome" would identify "Michael Fassbender" as the entity and "positive" as the sentiment. This chapter focuses on sentiment analysis in documents, since this is the analysis applied by FEAT.

The first step in performing sentiment analysis is determining features that are then represented in a structure known as a feature vector. Features of text include term presence or term frequency, parts of speech (POS), opinion words and phrases, and negations [16,20]. Terms are specific words or n-grams. An n-gram is a sequence of n words that appear consecutively in the document. Term presence may be represented in binary (present or not present), while frequency includes a count of the number of times the term appears in the text. Typically, POS features are adjectives since these provide a strong indicator of the author's opinion but may also include verbs, adverbs, and nouns. Opinions words and phrases are features that are commonly used to express positive or negative sentiment. For example, words such as "beautiful," "happy," and "good" provide a positive sentiment while words such as "ugly," "sad," and "bad" provide a negative sentiment. The appearance of a negation is also important to note since it changes the opinion of the text into the opposite sentiment. For example, a phrase such as "not good" is equivalent to "bad."

Feature selection methods reduce the original feature set by removing irrelevant features for text sentiment classification. These methods are either lexicon-based requiring human assistance, or statistical methods that are fully automatic. Lexicon-based approaches use sentiment lexicons to detect words that indicate an opinion. Statistical methods compute a score for each individual feature and then select the top ranked features using those scores.

These techniques can treat the document as a Bag of Words (BOW) or as a string that retains the original sequence of words. Because of ease, the document is typically treated as a BOW.

After features are selected, Sentiment Classification techniques are used to identify the sentiment expressed in the feature. These techniques can be categorized into two general approaches, Machine Learning or Lexicon-based. The Machine Learning Approach relies on traditional Machine Language algorithms that classify text by using syntactic and/or linguistic features. Supervised ML algorithms are provided a large set of training records where each record is given a specific label. The algorithm then infers information about the records that allows it to label new data. For example, a collection of movie reviews that have already been labeled as positive, negative, or neutral can then be used to train a classifier for classifying unlabeled movie reviews.

Another approach is to automatically label text per emoticons present in the text [9,18]. An emoticon such as **:)** indicates the text contains positive sentiment; an emoticon such as **:(** indicates negative sentiment. This technique is particularly useful for labeled text on social media sites such as Twitter where the use of emoticons is ubiquitous. The labeled text can then be used to train the classifier.

Unsupervised ML methods are used when training records are not available. These methods seek to derive training data in an unsupervised manner. For example, He et al. [10] used an unsupervised ML method to derive common categories from posts and tweets on Domino's, Papa John's, and Pizza Hut's Facebook and Twitter pages. Some common themes for the Facebook posts are pictures, questions, contest and games information, company activities, thank you, and promotions.

The Lexicon-based Approach relies on a sentiment lexicon, which is a collection of sentiment terms. There are two methods for building the sentiment lexicon: dictionary-based and corpus-based. The dictionary-based method starts with a seed list of opinion words that are collected manually. Next, the lexicon grows by using a dictionary or thesaurus to add synonyms and antonyms. After this process is complete, the lexicon is scanned manually to remove errors.

The problem with the dictionary-based approach is that the derived lexicon will not necessarily contain domain specific terms. The corpus-based approach solves this problem by finding opinion words that are specific to the domain. The approach uses a seed list of opinion words along with syntactic constraints to grow the list by processing a large body of text from the domain (a corpus). For example, the constraints could involve connectives like "and," "or," and "but." "And" would cause an adjective on one side of the "and" to be added if the adjective on the other side is already present in the list of opinion words. For example, if the corpus contains the phrase "insightful and beautiful," the opinion word "insightful" would be added with a positive orientation if the word "beautiful" was already present in the lexicon and labeled positive. This idea is called sentiment consistency. Similarly, a phrase such as "beautiful, but hateful" would cause "hateful" to be added with a negative orientation.

Hybrid methods for sentiment classification, combining both machine learning and lexicon-based approaches, are very common. In combining them, the sentiment lexicon built from the lexicon-classifier is used to label the training data that are subsequently used in the machine learning approach. Go et al. [9] found the hybrid approach performs better than a lexicon-based approach when executed on a set of IMDb movie reviews. The hybrid

approach performed only slightly worse than the machine learning approach and was much more convenient to use given that training data was not needed.

The sentiment classification approach used in FEAT is based upon the Naïve Bayes model, which is a supervised machine learning approach known for its simplicity and surprisingly good performance. The Naïve Bayes algorithm is based on two ideas: Bayes Theorem, and the naive assumption that the presence of a specific feature is independent of other features. The Naïve Bayes algorithm treats the text as a Bag of Words; in other words, grammar is discarded and word order is not considered.

Bayes' Theorem describes the probability of an event as based on conditions related to the event. Bayes' theorem is expressed in the following equation [16]:

$$P(label|features) = \frac{P(features|label)P(label)}{P(features)} \quad 3.1$$

where *P(label|features)* is the probability of having the label given that set of features. *P(label)* is the probability that any feature set is given that label. *P(features)* is the probability that a given feature set has occurred. *P(features|label)* is the probability that a given feature set is being classified with that label. For example, to calculate the probability that a review that contains the word "dislike" should be labeled negative, we would use the known negative reviews to calculate *P(dislike|negative)*, which is the probability that "dislike" appears among the negative reviews. This would be divided by *P(dislike)* which is the number of times the feature "dislike" appears among all extracted features from all labeled reviews. Finally, that result would be multiplied by *P(negative)* which is the number of negative reviews among all labeled reviews. The same calculations would be performed for the feature "dislike" and the label "positive" and the review would be labeled according to the probability that is greater.

Given the naive assumption that all features are independent, the equation can be rewritten as follows:

$$P(label|features) = \frac{P(label) * P(f1|label) * ... * P(fn|label)}{P(features)} \quad 3.2$$

Naive Bayes tends to overestimate the probability of a selected label; however, its classification decisions are quite good [24]. Thus, when used only to make the decision and not to accurately predict the actual probabilities, the model is accurate. In addition, the efficiency of the technique is useful when large amounts of data need to be categorized such as the data available on a social media site.

3.2 Word Frequency and Word Cloud

Word frequency analysis tools provide a list and the frequency of all the words in a text. Additional analysis can be performed to determine if some words are used significantly more in some parts of a document than in others. Further, words can be categorized, for example, "guns," "violence," "deaths," "2nd amendment," and these categories can be used to identify the topic of a document [14].

Word frequency analysis is one of the most common analyses performed on text and there are several tools available on the Internet for finding word or phrase frequencies. Some packages make use of synonym lists that allow the words to be categorized. For example, the Linguistic Inquiry and Word Count (LIWC) dictionary maps the word set {ashes, burial*, buried, bury, casket*, cemet*, coffin*, cremat*, dead death*, decay*, decease*, deteriorat*, die, died, dies, drown*, dying fatal, funeral*, grave*, grief, griev*, kill*, mortal*, mourn*, murder*, suicid*, terminat*} to the category death. The * in the set represents a wildcard; words that begin with the prefix will match the category. For example, graveyard matches grave.

Some tools lemmatize before the analysis. Lemmatization removes the grammatical structure leaving only the stem so that words are then counted as identical when they share a stem. For example, a tool that performs lemmatization would count “dying” and “die”as the same word. Additional preprocessing of text includes lowercasing words and possibly removing stop words. Stops words are common words such as “a,” “an,” “the,” “are,” etc. that do not contribute to the meaning of the document. There is no single list of stop words that is used by all text processing toolkits but multiple lists can be found on the Internet in various languages.

Word frequency analysis is useful in finding plagiarism, categorizing documents, and automated text summarizations. In addition, word frequency analysis can be used to establish the author’s writing “signature” that includes the use of words, phrases, slang, and jargon. Signatures can then be compared to identify an author. For example, this type of analysis was performed by computer scientists to identify J.K. Rowling as the author of “The Cuckoo's Calling” [12].

Word clouds display words with top frequencies in an amorphous image (cloud) or in an image with some meaning such as an elephant or donkey. Typically, words with larger frequencies are displayed using larger text sizes than words with lower frequencies. Word clouds allow us to quickly see the most significant topics in a document. For example, the word clouds from Obama's 2012 and 2014 State of the Union addresses both prominently display “America” and “American,” but the words “work” and “help” are much more prominent in the 2014 word cloud and thus, were more significant in that address [4]. Word clouds can also be useful for businesses. For example, a word cloud can be used to display

customer feedback, which can enable a business to quickly see what customers like and what they don't like.

3.3 Readability Analysis

Readability refers to the ease at which written text can be read and understood. Since the early twentieth century, hundreds of formulas have been designed to estimate the difficulty of text for readers. In recent years, due to technological innovations, there has been renewed interest in readability research. Computer Scientists, Linguists, Cognitive Scientists, and Educators have made strides in developing new methods for measuring readability of texts for various populations.

Although hundreds of formulas exist for computing readability, they all incorporate the same components including sentence length, word length, word frequency, and the length of paragraphs [4]. A passage with shorter sentences, smaller words, frequently used words, and short paragraphs is considered more readable than one that has long sentences, complex words, rarely used words, and long paragraphs. Beyond these easily measured components, passage cohesiveness can play a role in readability. A paragraph that consists of sentences that seamlessly move from one point to another is more readable than one that contains gaps which need to be filled in by the reader. Measuring reader comprehension of a text and comparing that measurement to what was predicted by the formula is the standard method for determining the validity of the formula. Critics would argue that there is no single method for calculating readability of a text and factors like style, grammar, and background knowledge are also critical but not measured.

Despite these shortcomings, readability scoring is widely performed. It is frequently used to determine grade appropriate texts for students. In addition, authors use readability

scoring to ensure their writing is accessible by the target audience. For example, the authors of medical pamphlets are interested in producing writing that is accessible to 7th and 8th graders, which is the average reading level of American adults.

The three readability formulas used in this thesis are Flesch-Kincaid, Gunning Fog Index, and Coleman Liau. Each of these produce output that approximates the U.S. grade level thought necessary to comprehend the input text.

Flesch-Kincaid Readability Formula uses the following equation to find the grade level of the text:

$$\text{FK} = 0.39\left(\frac{total\ words}{total\ sentences}\right) + 11.8\left(\frac{total\ syllables}{total\ words}\right) - 15.59 \qquad 3.3$$

This formula emphasizes sentence length over word length. In fact, large scores can be obtained with long sentences containing one-letter words.

The Gunning Fog Index (FOG) calculates readability as follows:

$$FOG\ index = 0.4\left[\left(\frac{total\ words}{total\ sentences}\right) + 100\ \left(\frac{hard\ words}{total\ sentence}\right)\right] \qquad 3.4$$

The FOG index deems "hard" words to be those with three or more syllables that are not proper nouns, jargon, or compound words. The FOG index is generally recognized as a useful measure of hard to read text, but it is limited in that it assumes that multi-syllable words are "hard" words, which is not necessarily the case.

The Coleman-Liau index is calculated using this formula:

$$CL\ index = 0.0588\ \frac{letters}{100words} - 0.296\ \frac{sentences}{100words} - 15.8 \qquad 3.5$$

Unlike syllable-based readability formulas, such as the FOG and SMOG index, the Coleman-Liau index does not require the character content of words to be analyzed. Only word and

sentence boundaries need to be detected, which means the index can be calculated in conjunction with text scanning.

In this thesis, we use readability assessment to score a text author's writing ability and indirectly, their US grade level. Studies indicate that reading and writing skills are complementary. Thus, applying a readability score to an author's writing provides an indirect measure of their level of literacy. One of the challenges of applying these formulas is the need to have a sufficiently large amount of text. If the amount of text is too small, the formula can produce results that are not within the standard grade-level range (1-13).

Chapter 4 - Methodology

The FEAT toolkit provides software that extracts data from a Facebook page, stores the data in a MySQL database, and allows the data to be viewed via a web front end. Figure 4.1 shows the complete process from data extraction to data visualization. This chapter discusses the components of the toolkit in detail. Section 4.1 discusses the FEAT Extractor that uses the Facebook API to extract data from a Facebook page and store data in a MySQL database. Section 4.2 describes the tools available in the FEAT Analyzer toolkit and their implementation. Finally, the FEAT Visualizer, which is used to display graphs on the website, is described in Section 4.3.

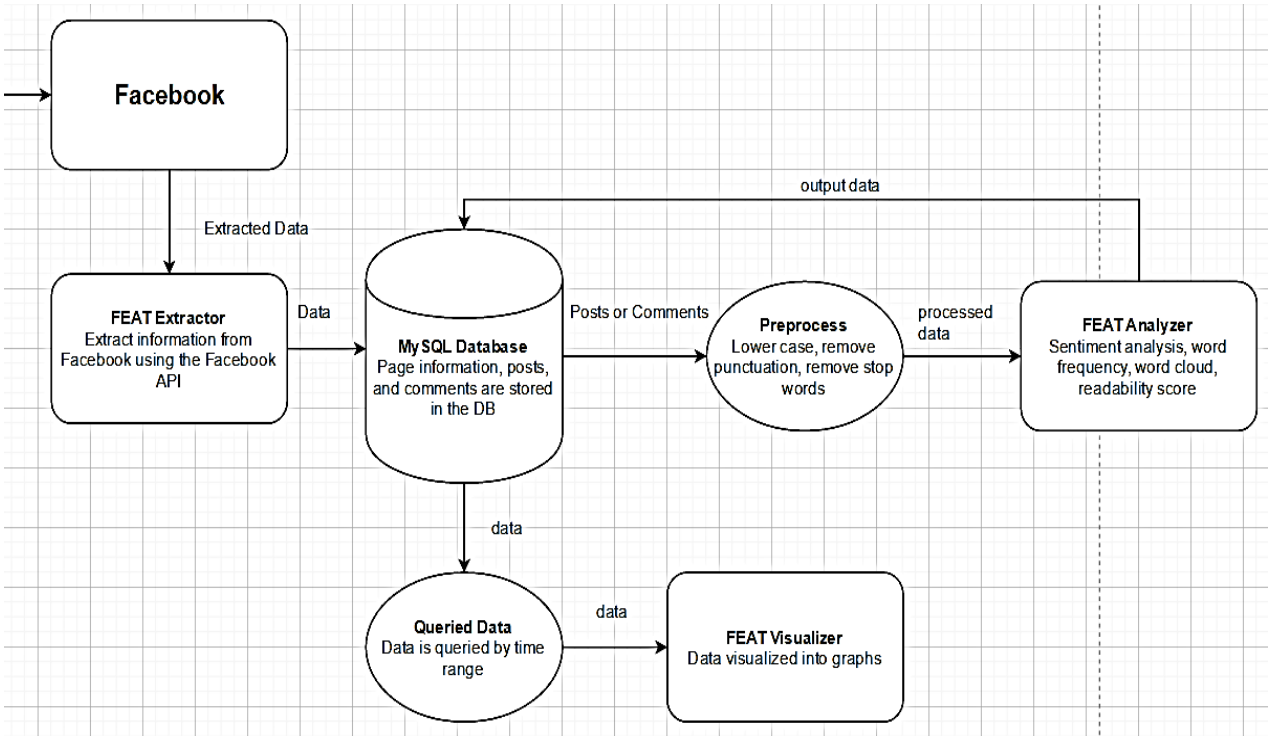


Figure 4.1: Overview of the FEAT toolkit

4.1 FEAT Extractor and Database Storage

Facebook provides three APIs: Atlas, Graph, and Marketing. The Graph API, which is the API used by FEAT, is a low-level HTTP-based API that is used to perform a variety of tasks such as querying data and posting new content. The Graph API represents Facebook data with nodes, edges, and fields. Each node represents a page or a page component such as a user, photo, post, or comment. Each edge represents a connection between nodes such as the connections between a photo and the photo's comments. Fields contain information associated with a node; for example, some of the fields associated with a photo are the photo id, the album containing the photo, and the time the photo was published to Facebook.

Using the Graph API requires registering with the Facebook developer website to receive an access token that will provide temporary access to Facebook APIs. The access token is a string of letters and numbers that can be used by an application to make API calls. There are three types of access tokens. A *user token* is needed to read, modify, or write a specific person's Facebook data on their behalf. An *app access token* is needed to modify and read the app settings. A *page access tokens* provides permission to APIs that read, write, or modify the data belonging to a Facebook Page. The type of token used by FEAT is the page access token.

Most Facebook objects can be accessed using the Graph API via the object's id. For example, a Facebook page object can be accessed using a page id. An alternative way to obtain a page object is by a URL such as https://facebook.com/hillaryclinton. The URL lookup will return the object, its associated fields, including the page id. Other objects, such as posts, can be found via the edges connected to the page object.

The FEAT Extractor is made up of two programs, extractor.py and pfExtractor.py. extractor.py is the main program of the FEAT Extractor. The extractor.py program takes a Facebook username, such as "HillaryClinton" and builds a Facebook URL, such as https://facebook.com/hillaryclinton. Next, the program makes a GET request to obtain the object associated with that user's page. The program then gains access to the public information available on the page via the page object such as the number of shares, number of followers, and ids of posts. The post id can then be used to obtain each post node. The post node contains a post id, creation date, message, number of comments attached to the post, number of likes, and number of shares. These fields are stored in the database. The FEAT Extractor also uses the post node to find the list of associated comment ids. Each id is then used to obtain the comment node, the comment id, message, author id, comment creation date, and the number of likes is stored in the database.

The pfExtractor.py script is automatically executed at the end of each day to retrieve the number of followers of a page and store that value into the database. The script is provided a list of usernames of pages from which the number of followers is extracted. The toolkit is currently being used to extract information from the public Facebook pages of Donald Trump, Hillary Clinton, Bernie Sanders, Ted Cruz, and John Kasich. Daily counts of the number of followers of each page are then stored in the database. Data extracted from Facebook through the Graph API is stored in a MySQL database. Figure 4.2 illustrates the database schema for the FEAT MySQL database.

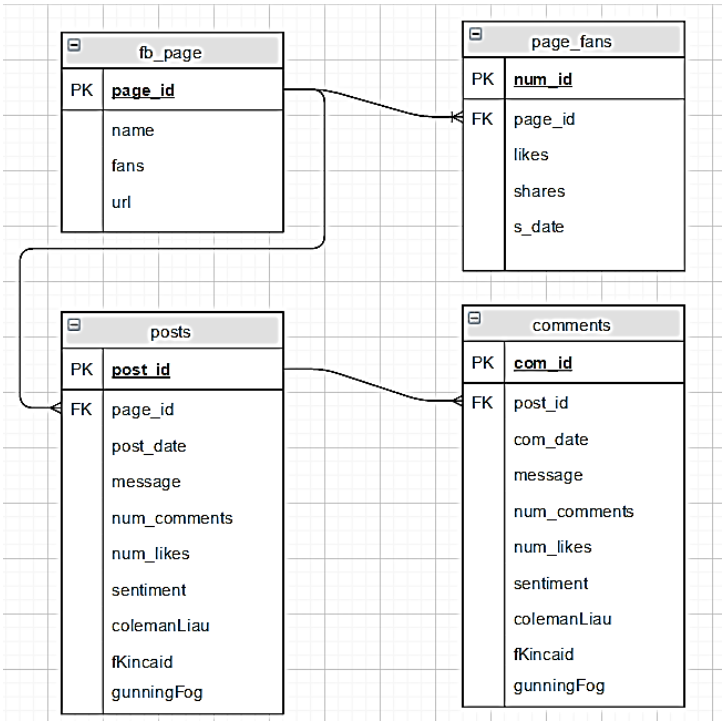


Figure 4.2: FEAT database schema

The fb_page table contains the page id, name, number of followers, and the URL of each candidate's Facebook page. The page_id is the primary key for the fb_page table. A primary key ensures that there are no two rows in the same table with the same page_id. The page_id is a foreign key in the other tables to provide a connection between a page and the content on the page. For example, the page_id in the posts and page_fans tables are the foreign keys that refer to the page_id in the fb_page table. The post_id is the primary key for the posts table. The comment_id is the primary key for the comments table. These keys can be seen in the table.

The sentiment and readability (colemanLiau, fKincaid, and gunningFog) score fields are initially left as null. These fields are set by running the appropriate python program. The readability score fields are set to a decimal number. Any score below 0 is set to 0, meaning that the text is considered kindergarten level. Scores 13 or above are considered college level.

The sentiment analyzer finds the sentiment polarity of posts and comments. The analyzer returns a label of either positive, negative, or neutral that is stored in the post or comment table.

Posts and comments may be queried from the database to find the top twenty-five words with the highest frequency. These are visualized in a word cloud or in a word frequency graph.

4.2 FEAT Analysis Tools

This section discusses information about the Analysis tools within the FEAT toolkit. These include tools to perform word frequency, generate word clouds, analyze the sentiment of the text, and calculate readability. The tools are based upon the Natural Language Toolkit (NLTK) platform (http://www.nltk.org/). NLTK is a leading platform for building Python programs to analyze language. NLTK provides a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

4.2.1 Word Frequency and Word Cloud

The word frequency toolkit can be applied to a set of comments or a set of posts within a specified time frame. The process of determining word frequencies is as follows:

1. The text is tokenized using the function word_tokenization() provided by NLTK, producing a list of all words that appear in the text.
2. All words in the list are lowercased.
3. Words that appear in the stop word list are removed. Punctuation is also removed.

The *nltk.corpus* package defines a collection of corpus reader classes that can be used to access various sets of corpora. Among those sets are lists of stop words that were used in step 3 above. In this analysis, the stop words used were gathered from xpo6.com and

MAXQDA. The NLTK FreqDist class is used to calculate the total number of word tokens distributed across the text. The FreqDist class encodes frequency distributions, which count the number of times that each outcome of an experiment occurs. For example, the frequency could be the number of times a word appears in a document. The FEAT code creates a FreqDist object, passing to the constructor the tokenized words. The constructor returns a FreqDist object that allows access to the frequencies via a dictionary; the words can be accessed as keys and the counts as the values. The FreqDict most_common method is used to return the 50 most frequently used words along with their counts. This word frequency functionality was implemented in view.py, which is later explained in Section 4.3.

Word clouds are a good way to summarize text and to visualize the top words with the highest frequencies in the text. The FEAT word cloud tool can be used to visualize either the comments or the posts on a page. The word cloud generator used by FEAT is open source and available on Github at https://github.com/amueller/word_cloud. This software has been used by other significant projects including for Reddit Cloud, which is a Reddit bot that generates word clouds for comments and user histories. This software is also used in a Twitter Word Cloud Bot that is automatically invoked by Twitter users via the hashtag #wordcloud; the #wordcloud hashtag will cause the bot to create a word cloud from the user's tweets.

The FEAT word cloud tool, runWordCloud.py, connects to the database and queries it for either posts or comments within a specified data range. The tool then creates a file that contains words separated by commas (a csv file), which is the required input for the word cloud generator. The word cloud generator performs word frequency analysis to find and

display the most frequently used words. The word cloud generator can also be passed an image that is used to display the word cloud in the shape of the image as shown in Figure 4.3.

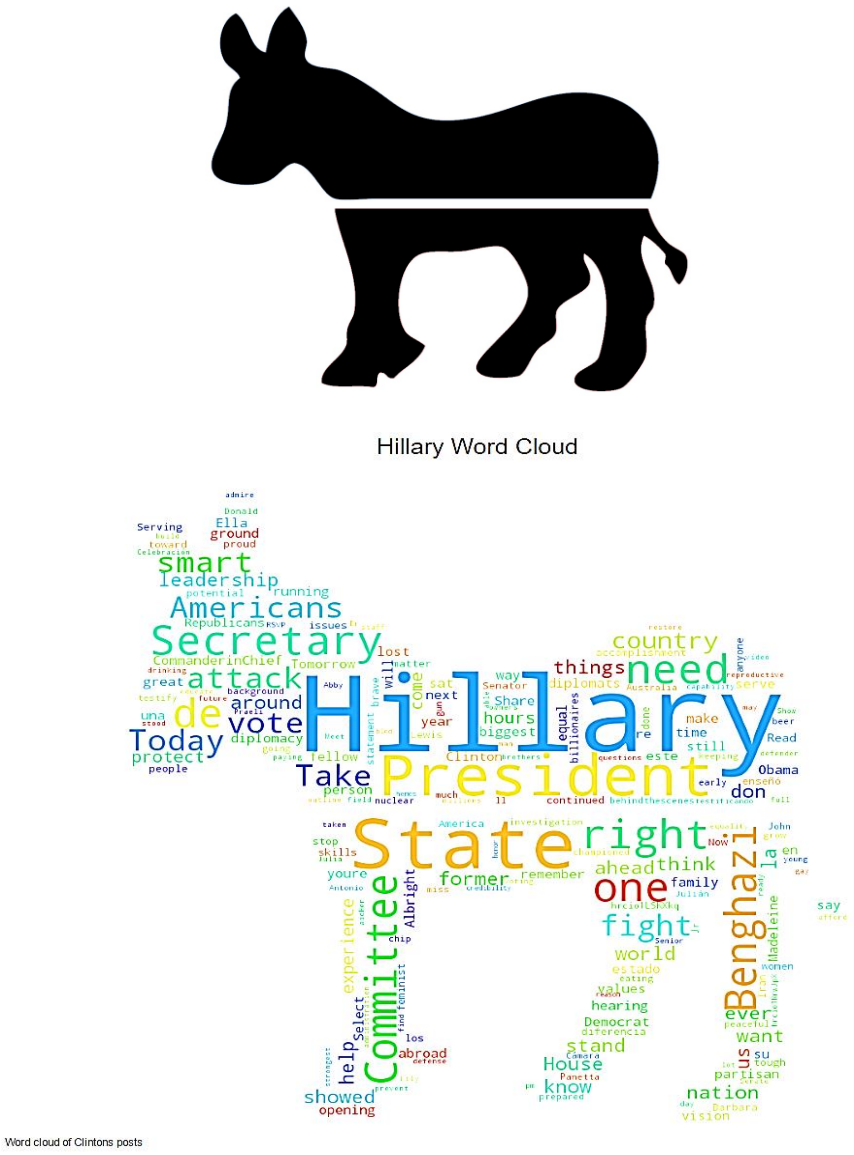


Figure 4.3: Donkey shape image and an example word cloud image

4.2.2 Readability Analysis

The FEAT readability tool performs readability analysis on either posts or comments. The program calculates a readability score on individual posts or comments using the Python Readability API. Three formulas are used: Flesch-Kincaid Grade Level, Gunning Fog Index, and Coleman Liau Index. The readability score fields in the database are named

colemanLiau, fKincaid, and gunningFog. These are set by using the Coleman Liau Index, Flesch Kincaid Grade Level, and Gunning Fog Index formulas, respectively. The result of each formula is a grade level that corresponds to the text readability. The view.py script explained in Section 4.3 uses the three fields to produce an average grade level.

4.2.3 Sentiment Analysis

The FEAT sentiment analyzer tools uses the sentiment analysis API provided by text-processing.com, which in turns uses tools from the NLTK to analyze text. The analyzer can be applied to either posts or comments. The http://text-processing.com API is an HTTP web service for text mining and natural language processing; the service returns a JSON object with the attributes label and probability. The label attribute represents the result of the sentiment analysis of the text by listing the sentiment as "pos" for positive, "neutral," or "neg" for negative sentiment. The probability attribute shows the probability of the text having positive, neutral, or negative sentiment values. The probabilities of the text having a positive or negative sentiment are between 0 and 1 and will add up to equal 1. The text is considered neutral if the probability of it being neutral is greater than 0.5. Otherwise, the result will be either "pos" or "neg."

The http://text-processing.com sentiment analyzer API uses the Naïve Bayes model provided by NLTK to perform text classification. Details of the Naïve Bayes model can be found in Chapter 3.1. Movie reviews and Twitter sentiment from the dataset provided by Bo Pang and Lillian Lee were used to train the classifier. The NLTK-Trainer is used to train NLTK based models, evaluate models against a corpus, and analyze a corpus. With the NLTK-Trainer, one may also use a pre-trained classifier to analyze the text. The drawback to using this sentiment analysis API is that there is a limit of 80,000 characters per text and each

method is throttled to 1000 calls per day per IP address. However, a subscription allows 45,000 calls per month and $0.01 per extra call.

After the text is analyzed, the label is stored in the database under the sentiment attribute of the respective row in either the posts or comments table. The view.py software can then be used to provide a visualization of the sentiments.

4.3 The Website

The visualization component uses the Django framework and is hosted on PythonAnywhere (https://www.pythonanywhere.com/). PythonAnywhere is an online Integrated Development Environment (IDE) that provides web-hosting services. In addition to hosting the website, PythonAnywhere offers a bash command-line interface along with a code editor. PythonAnywhere allows for the creation of an app with the use of various popular frameworks such as Django and Flask. In addition, PythonAnywhere permits usage of MySQL, SQLite, and Postgres databases.

The Django framework is designed to make common web-development tasks fast and easy. Django comes with an object-relation mapper in which the database layout is described in Python code that maps to the database. In addition, data-model syntax is used to represent each table in the database. In the data-model syntax, all attributes of the table are represented as fields in a model. The model specifies the primary and foreign keys along with the data type for each field. Django also allows for the configuration of URLs for the website, permitting the removal of endings such as ".php" and ".asp." All links pertaining to the website such as the front page and the administrator page are mapped in urls.py.

The two main components of the website are view.py and chartController.js. The view is responsible for returning an HttpResponse object that contains the content for the

requested page. The view retrieves the data for the page content along with any arguments, loads the specified template, and renders the template with the content data. The responsibilities of view.py includes:

- Set control data for the homepage
- Query for the post/comments from the database that needs to compute the word frequency
- Execute the word cloud program if the needed word cloud image does not exist
- Query the database for the readability score and get the average grade level
- Query the database for the sentiment polarity and find the percentage for each grade level
- Build the context data that is returned to the home page and passed to ChartController.js

The view.py script handles querying the database for the sentiment values, readability scores, and constructs the context data that is returned to the home page. The home page then passes the context data as arguments to the chartController.js script that handles the visualization of the graphs. The view.py script passes to a chartController.js script the graph type, candidates, title, descriptions, data such as counts for the total of positive, neutral, and negative messages, and labels for each data value.

The chartController.js script builds the sentiment, readability, or frequencies graph when the respective button is clicked. It uses the context data that was passed in as the arguments to create the data object that Chart.js uses to create the graph. If the graph type is specified as “word,” chartController.js first locates the word cloud image that was produced

by the word cloud script by forming the URL of the image location. Finally, it replaces the chart object with the word cloud image. The description is then placed beneath the chart. For the other graphs like the bar graph and pie chart, chartController.js creates a data object with the specified graph type, respective dataset and labels that were passed in the arguments. This data object is use by Chart.js to produce the specified graph.

One of the most powerful parts of the Django framework is the automatic admin interface which reads metadata from the database to allow administrators to modify the content of the website. The admin interface is customizable and can be activated or deactivated depending on the administrator. Figure 4.4 shows the administrator front page. The administrator may modify groups, users' information, and change the graphs that are viewed on the website's front page. By clicking the add button, the administrator is shown the Add Graph Form in Figure 4.5. This form allows an administrator to specify the graph title, candidate, graph type, description, and date range. This form is created by importing the my_app.Graph model from the database. The candidate and graph type are provided with a drop-down list with choices that are defined in the model. As soon as the new/modified graph is saved, a graph label is added to the side bar where it can be clicked to display the respective graph. Chapter 5 displays screenshots of the graphs that can be built using the FEAT toolkit and compares these results to similar work.

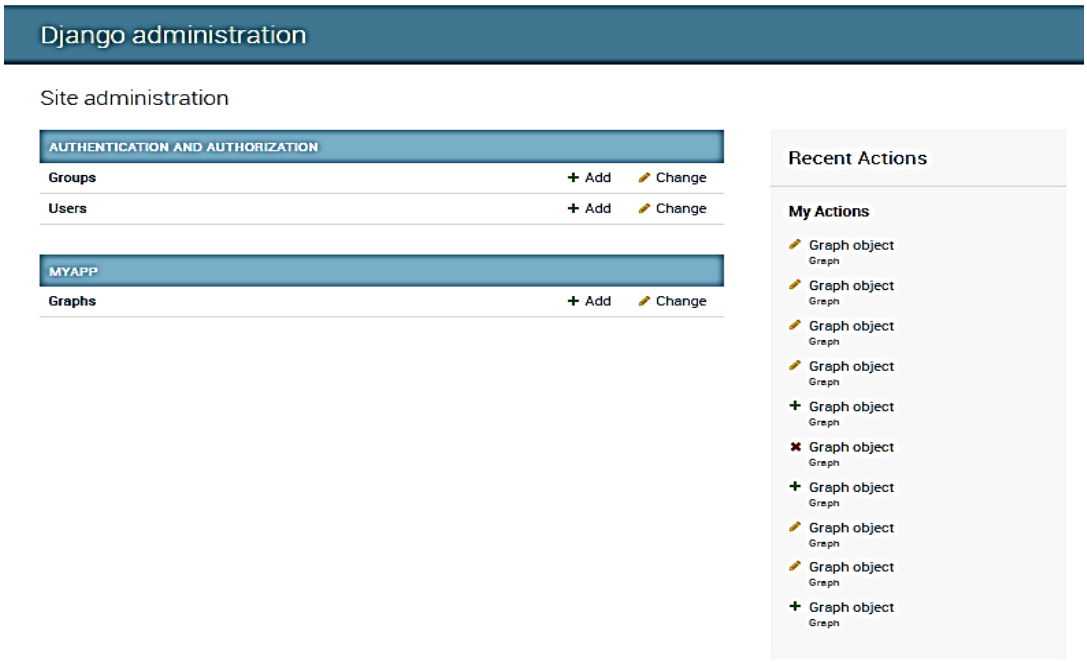


Figure 4.4: FEAT administrator main page

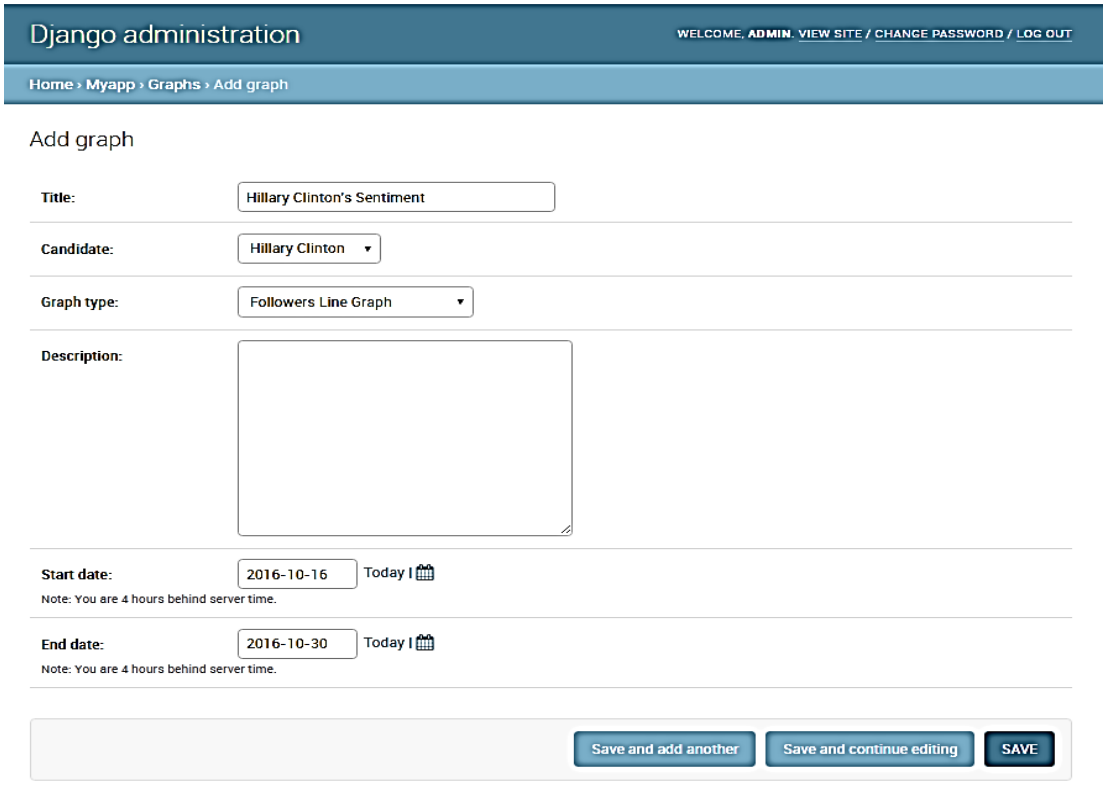


Figure 4.5: FEAT administrator add graph page

Chapter 5 - Results

Respected election forecasting sites, including The New York Times UpShot, Nate Silver's FiveThirtyEight, Princeton's Election Consortium and the Cook Political Report, incorrectly predicted that Hillary Clinton would win the 2016 US election. In the days that followed the election, many data scientists tried to figure out the reason why their prediction models failed. Some suggested reasons for the bad polling results range from voters hesitant to admit their support of Trump to pollsters under sampling the non college-educated crowd that voted for Trump. However, analysts of social media claim that social media sites predicted a win for Trump months ago [21]. This chapter discusses the results of using FEAT to extract and analyze data from Hillary Clinton's and Donald Trump's Facebook pages. Although the tool was not used to predict a Trump win, there are indications in the data collected that make a Trump win less surprising.

Section 5.1 discusses line graphs that show the counts of followers throughout the year. Section 5.2 discusses word frequency graphs and word clouds. The word frequency program was used to calculate the top 50 words used by commenters on Election Day, November 8, 2016. The word cloud software was used to display the top fifty most commonly used words in candidate posts. Section 5.3 discusses graphs displaying the sentiment polarity of candidate posts. This information is displayed in a bar graph that shows the percentage of positive, neutral, and negative posts per week within the specified date range. Section 5.4 discusses the readability of the posts and comments written on the Facebook page.

5.1 Followers

Social media analysts recognized that Trump received more reactions to his social media presence than his competitor did even though Clinton posted on her social media sites at the same rate [22]. The graphs of followers also indicate that Trump was more effective in engaging voters online. Figure 5.1 and 5.2 show the number of followers of the Clinton or Trump Facebook pages in the time range between September 8 and November 8. Even though the number of followers for both candidates increased at a steady rate, the number of Trump followers was significantly larger. On September 9th, Trump's Facebook page had twice the number of followers as Clinton's page (10 million versus 5 million). This gap narrowed somewhat as Election Day approached, but still on Election Day, Trump had almost 50% more followers than Clinton (12,000,000 versus 8,000,000).

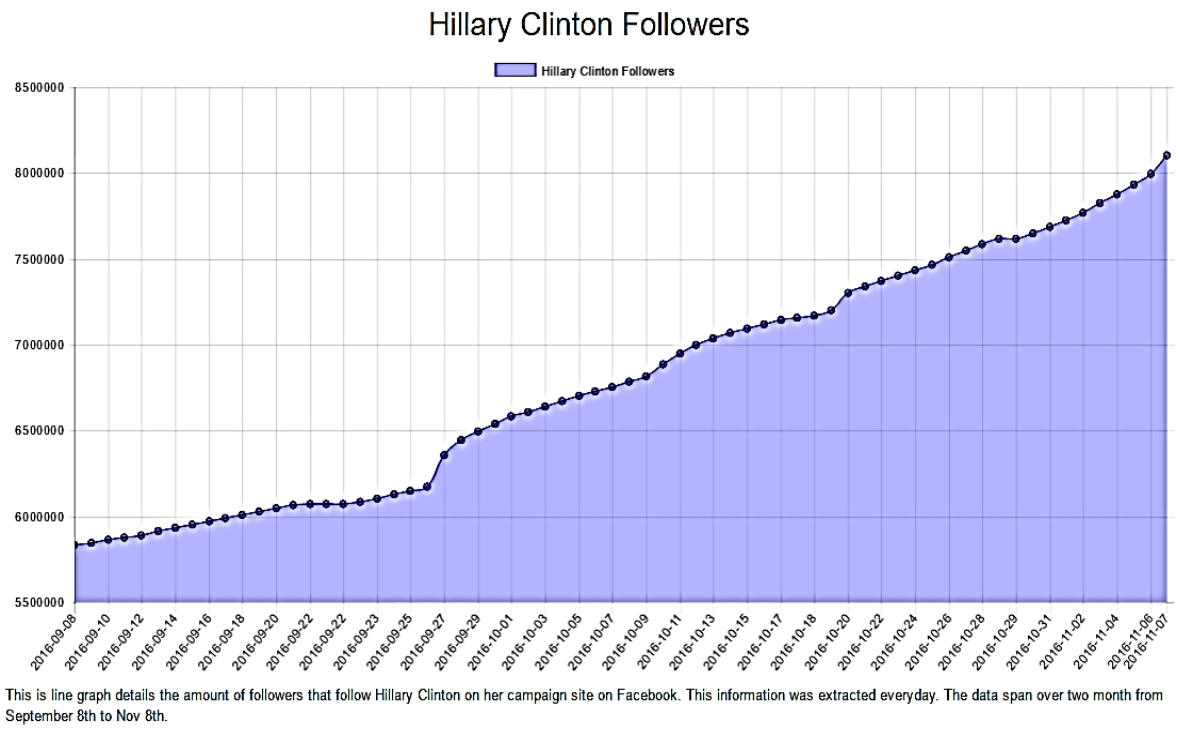


Figure 5.1: Clinton followers in 2016 election (September – November)

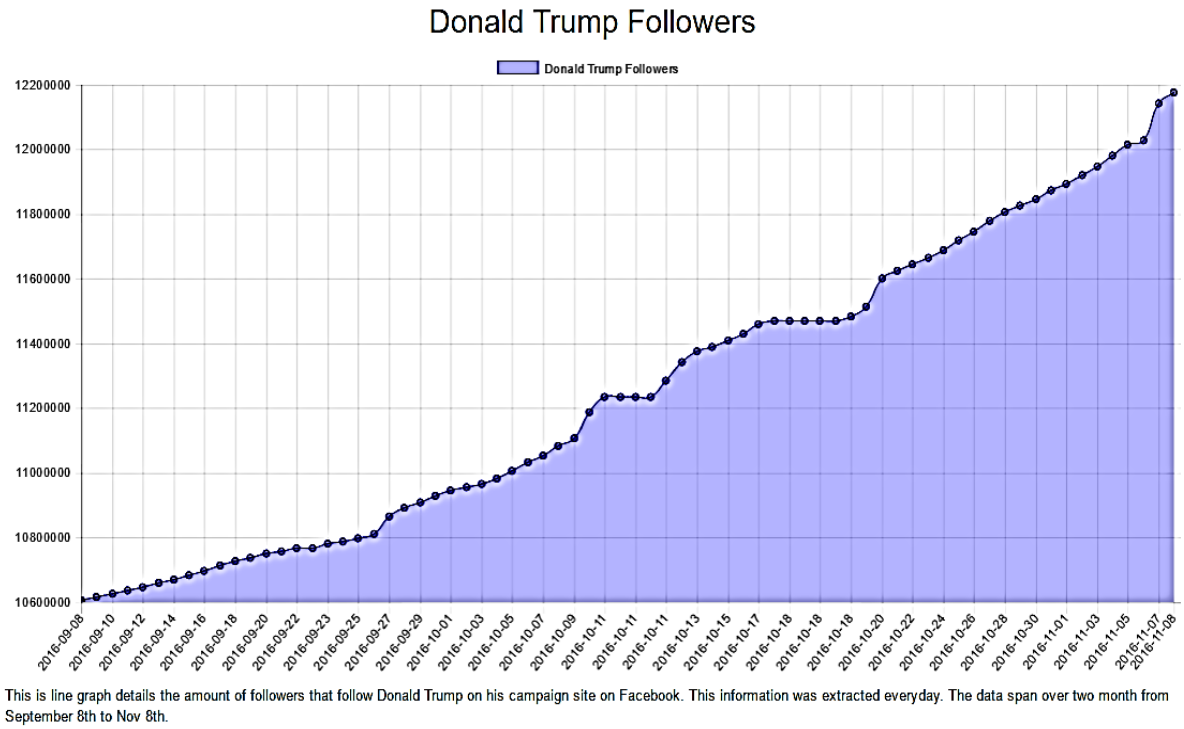


Figure 5.2: Trump followers in 2016 election (September – November)

5.2 Word Frequency and Word Cloud

Word clouds have become quite popular for expressing popular topics in a document or collection of documents. Word cloud software creates an image out of the 50 most commonly used words in a document where more frequently used words are displayed in a larger font than words less frequently used. Figures 5.3, 5.4, and 5.5 display the word clouds formed from posts on Clinton's Facebook page in the month of February, the month of May, and November 1-8, 2016. Figures 5.6, 5.7, and 5.8 contain word clouds from Trump posts during those same dates. The FEAT word cloud software was used to build the word clouds for November. The word clouds for February and May were created before the FEAT word cloud software was added to the toolkit. Thus, these word clouds were generated by extracting the data from the database, creating a csv file, and copying the csv file contents into the online word cloud generating tool known as Wordle (http://www.wordle.com).

As noted by Benjamin [3], democratic candidate pages tend to frequently use the name of the candidate and this is apparent in the February and May word clouds where "Hillary" is displayed in very large font. In addition, the clouds identify topics that were important to the Clinton campaign that are not noted in the word clouds of Trump. These topics include "families," "women," and "need." Some words were used more often as the election approached, for instance, the words "president" and "America." The word "vote" is displayed prominently in the word cloud built from November posts.

Figure 5.3:Word cloud of Clinton's posts (February 2016)

Figure 5.4: Word cloud of Clinton's posts (May 2016)

Figure 5.5: Word cloud of Clinton's posts (November 2016)

The word clouds built from Trump posts reflect his personality and his campaign slogans. “Great” and “MakeAmericaGreatAgain” appear prominently. In addition, words related to Trump's life outside politics also appear in the earlier clouds, such as “apprentice,” “tower,” “golf,” and “celebrity,” but disappear altogether in the November word cloud. Similar to the Clinton results, the word “vote” is displayed in large font in the November cloud; however, the words “great,” “American,” “live,” and “make” are in larger font.

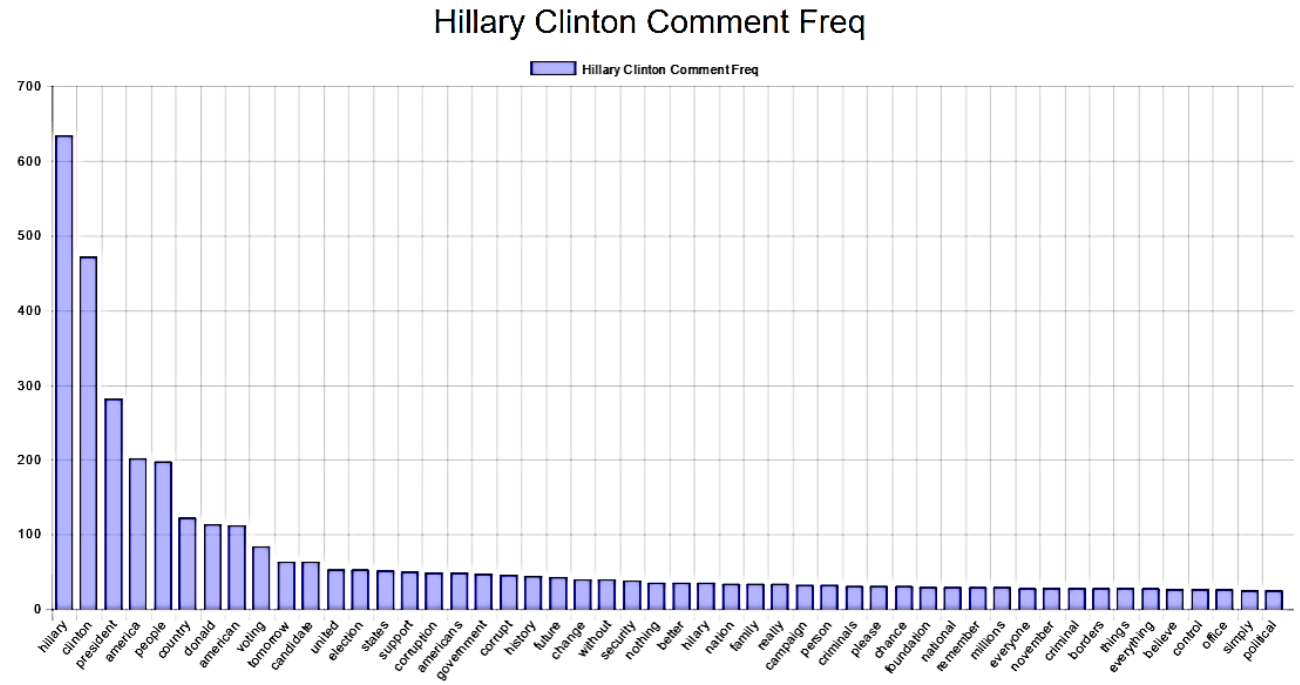
Figure 5.6: Word cloud of Trump’s posts (February 2016)

Figure 5.7: Word cloud of Trump's posts (February 2016)

Figure 5.8: Word cloud of Trump's posts (November 2016)

Other researchers have also gained insight into candidate's views by creating word clouds. For example, Kumar et al. [13] created word clouds from the pages of Ted Cruz, Chris Christie, Ben Carson, Jeb Bush, Bernie Sanders, Martin O'Malley, Hillary Clinton, Marco Rubio, Rand Paul, Mike Huckabee, Lindsey Graham, Donald Trump, and Rick Santorum. A simple analysis of these word clouds reveals the issues of importance to the candidates. The democratic candidates are concerned about social issues thus "health," "care," "women," and "families" are prominently displayed. The GOP images prominently display "tax," "security," and "government."

Figures 5.9 and 5.10 display the word frequency distributions taken from Clinton's Facebook page and Trump's Facebook page, respectively. These graphs display the top 50 words extracted from comments made on those pages on Election Day, November 8, 2016. For both pages, the top words were "hillary" and "clinton," thus followers were commenting about Clinton on both Trump's Facebook page and Clinton's Facebook page. The name "donald" also appeared frequently on both pages. Since the word frequency graphs were built from data collected on Election Day, the words "voting," "election," and "tomorrow" appear frequently on both pages. It is likely that not all commenters on Hillary Clinton's page were supporters since "corruption" and "corrupt" also appeared in the frequency graph and Trump frequently referred to Clinton as corrupt.



This is the word frequencies bar chart that displayed the top 50 words used by commentators on Hillary Clinton Facebook Page. English and Spanish stop words are removed.The comments are analyzed for November 8.

Figure 5.9: Word frequency graph for comments on Clinton's page (November 8, 2016)

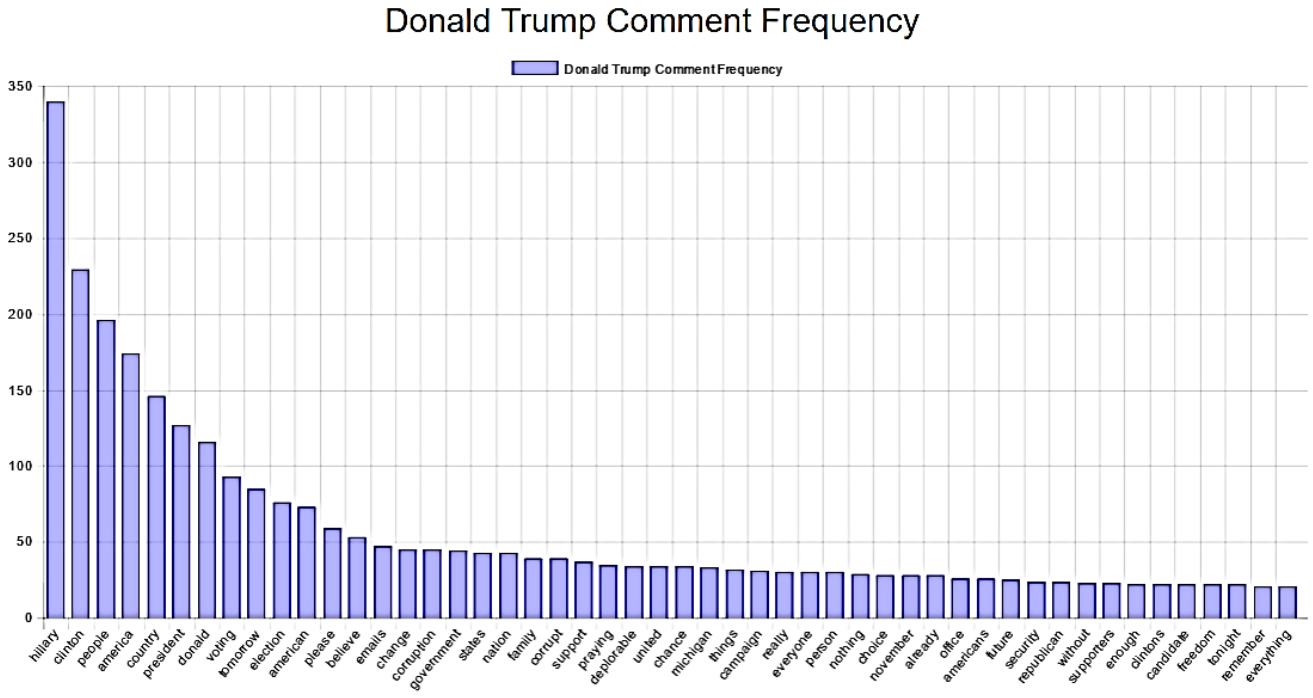


Figure 5.10: Word frequency graph for comments on Trump's page (November 8, 2016)

5.3 Sentiment Analysis

Sentiment analysis was performed on posts made by each candidate during the time range between September 9, 2016 and November 9, 2016. Each post was individually scored to determine the sentiment (positive, negative, or neutral) and these scores were grouped into weeks. Figure 5.11 displays the percentage of positive, negative, and neutral posts of Hillary Clinton during each week of the time range. Figure 5.12 displays the same type of graph based upon Trump's posts. Both graphs indicate that most of the posts were neutral and there were more positive posts than negative ones. Although both candidates' posts were more frequently positive than negative, the graphs indicate that a larger percentage of Trump's posts were positive, especially during the last week of the election.

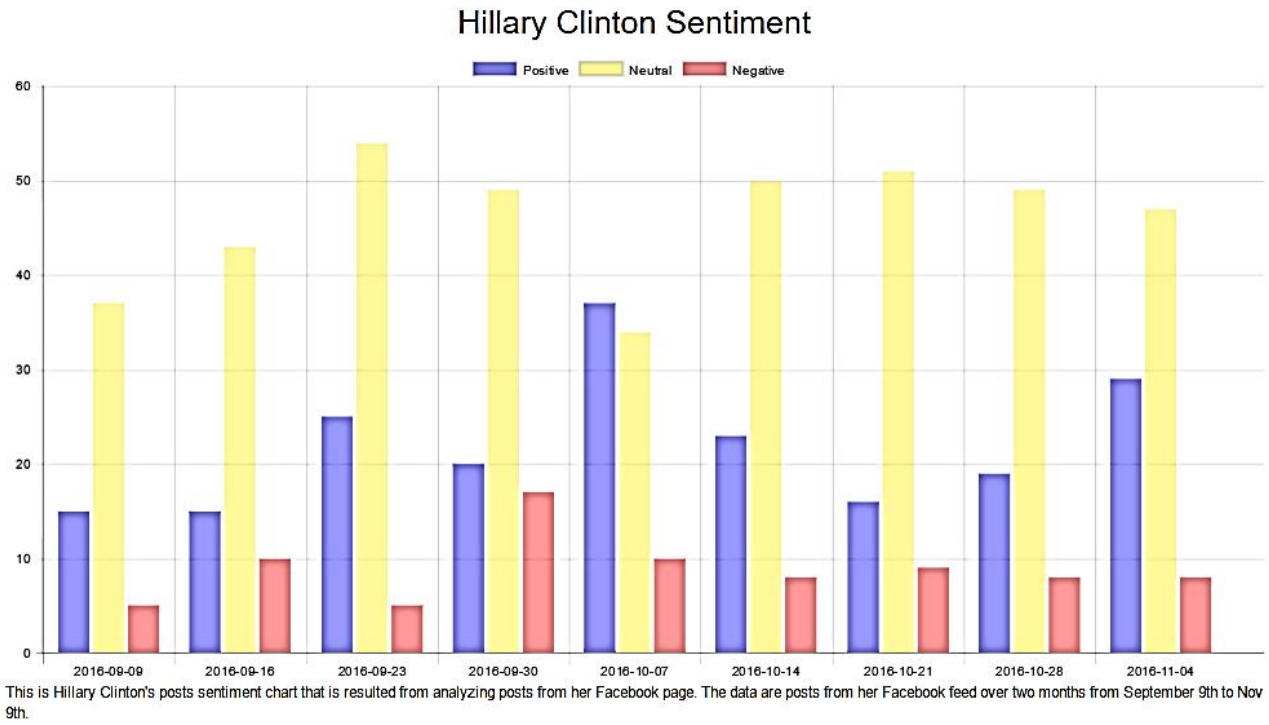


Figure 5.11: Sentiment analysis of Clinton's posts (September – November, 2016)

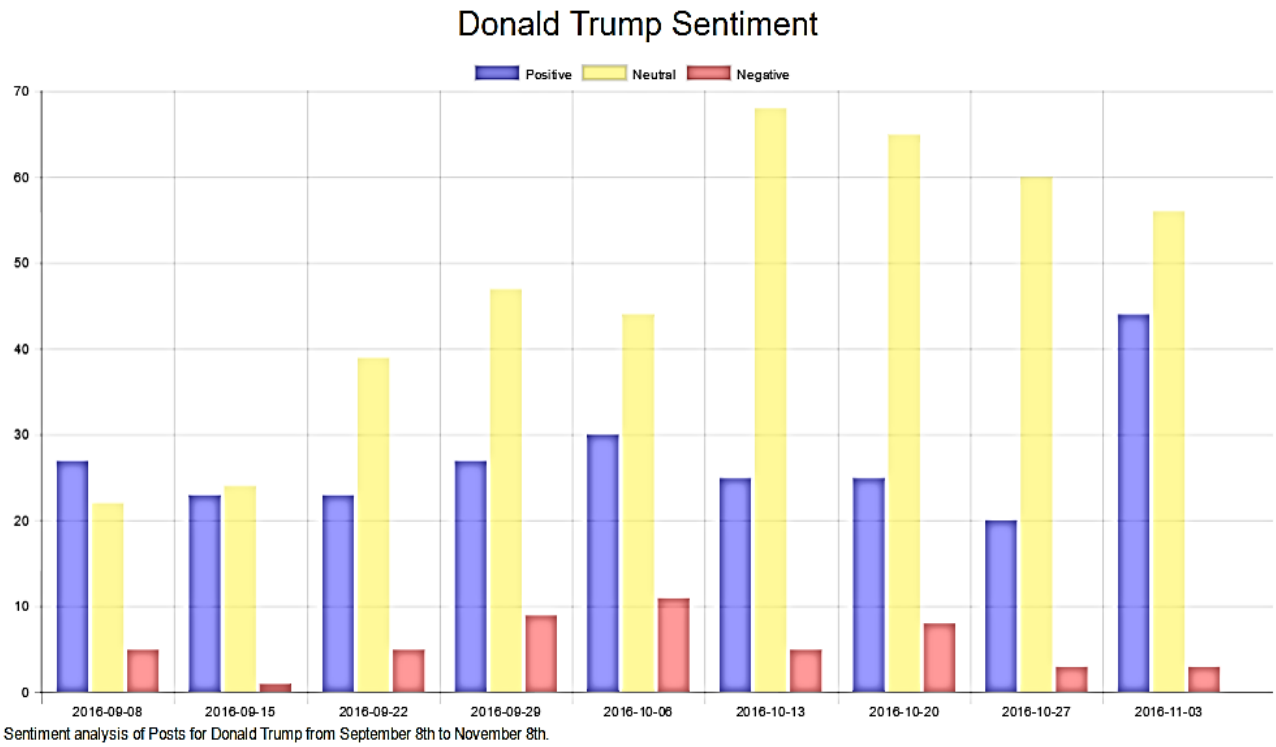
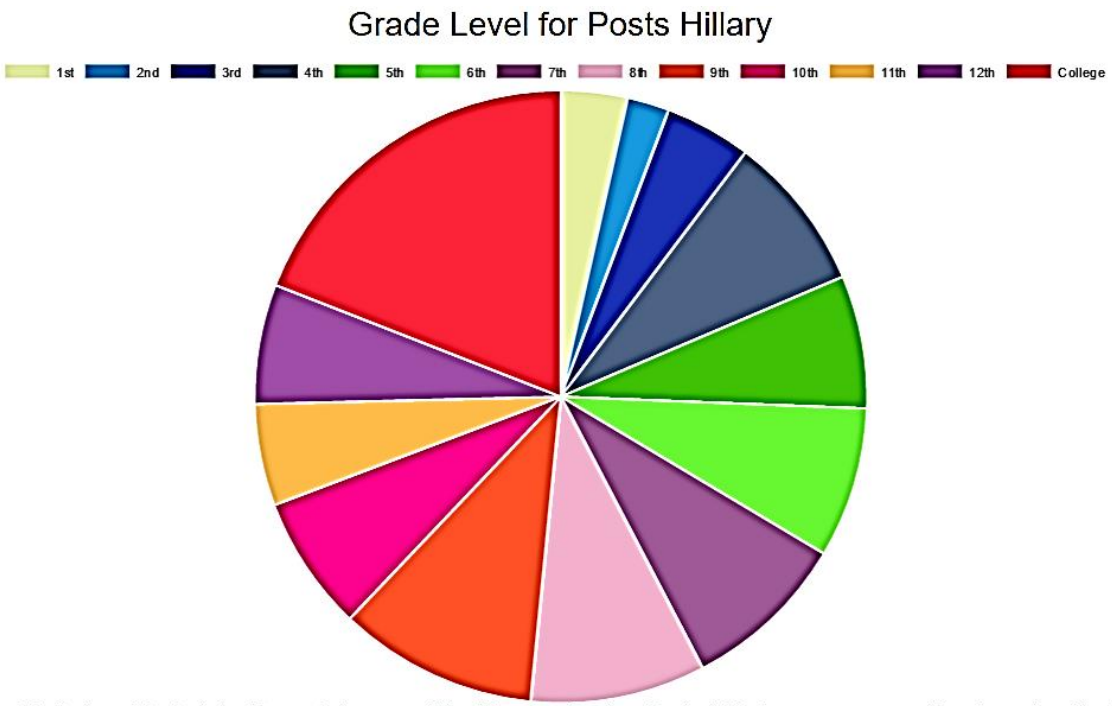


Figure 5.12: Sentiment analysis of Trump's posts (September – November, 2016)

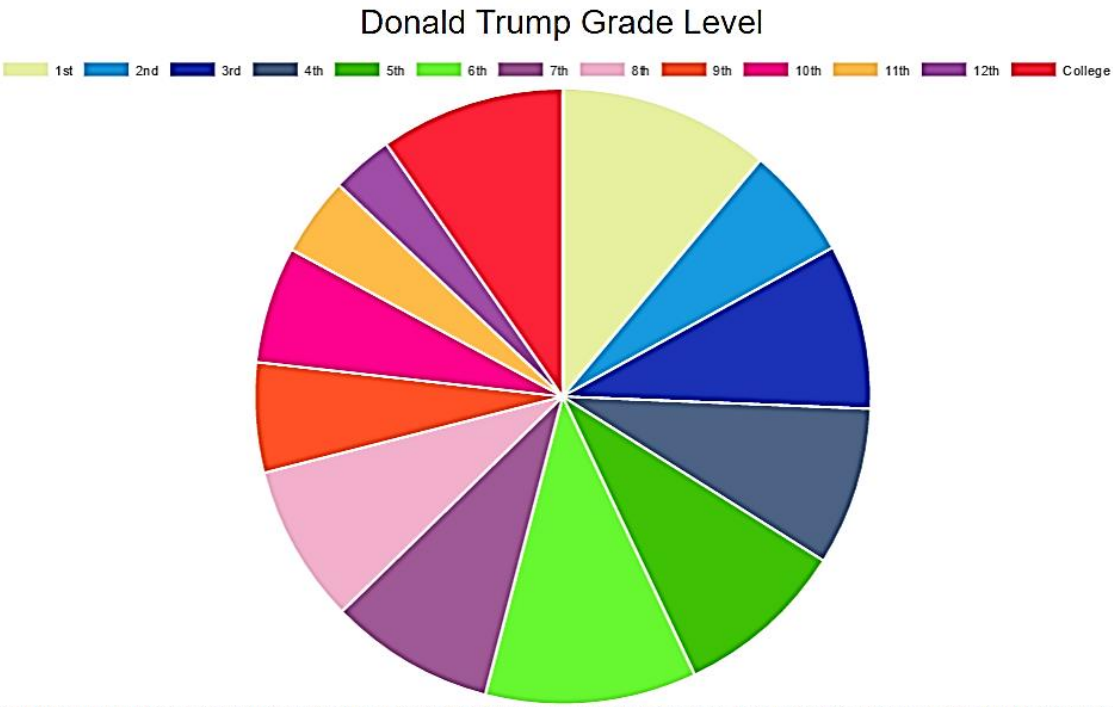
5.4 Readability

The readability score was calculated by using the formulas for Flesch-Kincaid Reading Level, Gunning Fog Index, and Coleman Liau Index and averaging those scores. Figures 5.13 and 5.14 display the reading grade level of the posts on Clinton's and Trump's Facebook pages, respectively. These posts were made by the candidates during the time range from August 1, 2016 to November 8, 2016. As can be seen in the pie charts, a much larger percentage of Clinton's posts scored at the college level compared to the posts of Trump. In addition, 75% of Trump's posts score at or below the 8th grade level compared to about 50% of Clinton's. What is unknown is whether the candidates were deliberately attempting to create posts targeting a specific grade level or whether these reflect the actual level of literacy of the candidates.



This is pie graph details the grade level calculated from analyzing posts on Hillary Clinton campaign site on Facebook. The data span over two month from August 1st to Nov 8th.

Figure 5.13: Readability scores on Clinton's posts (August – November, 2016)



This is pie graph details the grade level calculated from analyzing posts on Donald Trump campaign site on Facebook. The data span over two month from August 1st to Nov 8th.

Figure 5.14: Readability scores on Trump's posts (August – November, 2016)

The author in Kumar et. al [13] also performed a readability analysis on the Issue pages of presidential candidates using only the Flesch-Kincaid Reading Level formula. Similar to the results presented in Figures 5.13 and 5.14, the Clinton readability score is higher.

Chapter 6 - Conclusion and Future Work

The FEAT toolkit can be used to extract and analyze any public Facebook page. This thesis explained the design and implementation of FEAT and applied the toolkit to the pages of the presidential candidates, Hillary Clinton and Donald Trump. At present, the toolkit can be used to build graphs to display four different analyses: Sentiment, Word Frequency, Readability, and Word Cloud. The FEAT interface allows the user to select the candidate page, select the date range, and select the analysis to be performed. The software was designed so that additional analyses can be added to the toolkit without significant modifications.

Sections 6.1, 6.2, and 6.3 discuss limitations to the analysis components of the FEAT toolkit. Section 6.4 discusses future work that would improve the toolkit.

6.1 Word Frequency and Word Cloud

The Word Frequency and Word Cloud analyses depend upon software that removes "stop words." The Natural Language Toolkit provided a list of stop words that includes about 200 common English words, some Spanish words, and some punctuation. The list of stop words could be made larger. In addition to increasing the list of stop words, removing URLs and stemming could provide better results. For example, stemming would have eliminated the appearance of both words "corruption" and "corrupt" in Figure 5.9 since both of these words have the same root (corrupt).

6.2 Sentiment Analysis

The sentiment analysis API used the Naive Bayes algorithm to successfully classify the text into positive, neutral, and negative categories. There was, however, a large amount of neutral results. A variety of algorithms are available for performing sentiment analysis and it

is possible that other techniques may be more appropriate for performing sentiment analysis on the typically small amount of text available in a post.

6.3 Readability

Readability scores are limited in that they do not indicate whether the content of the measured document is understandable. In fact, it is possible to obtain artificially high readability scores by exploiting knowledge about the formulas used in the scoring. A common component in readability scoring is length of sentence where longer sentences lead to higher scores. Marco Rubio's page achieved the highest readability score among the 13 pages analyzed by [3] because of his frequent, but not always understandable, long sentences.

A related analysis that may provide more reliable results is Lexile analysis (http://www.lexile.com). Lexile analysis scores text based upon syntactic complexity (sentence length) and semantic difficulty (vocabulary). Words in the text are matched to those in a 650 million-word corpus to obtain values for computing the semantic difficulty.

6.4 Future Work

Recently, Facebook has enabled users to share reactions to posts that include love, haha, wow, sad, and angry in addition to the original “like” reaction. These reactions can permit posts and comments to be categorized according to the reactions, in addition to being categorized by the content. An interesting study would be to investigate a relationship between the reactions to a comment or post and the sentiment expressed by the post.

In addition, posts on social media not only include text but also images and videos. At present, FEAT is unable to analyze posts that contain only an image or a link to a video. When the FEAT analyzer performs sentiment analysis, a post without text is categorized as neutral. In addition, a readability score of these types of posts is 0. Although this readability

score is appropriate, sentiment analysis of video or the text within images could provide useful information. The main issue is how to interpret the image or generate text from the audio component of the video. Speech recognition software may be useful for this effort.

Bibliography

[1] Sitaram Asur and Bernardo A. Huberman. Predicting the Future With Social Media. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on*, Toronto, ON, 2010.

[2] Eytan Bakshy. The 2012 Election Day Through the Facebook Lens. https://www.facebook.com/notes/facebook-data-science/the-2012-election-day-through-the-facebook-lens/10151181043778859/. 2012.

[3] Rebekah George Benjamin. Reconstructing Readability: Recent Developments and Recommendations in the Analysis of Text Difficulty. *Educational Psychology Review.* 2011.

[4] Boost Labs. Word Clouds & the Value of Simple Visualizations. http://www.boostlabs.com/what-are-word-clouds-value-simple-visualizations/. 2016.

[5] Murphy Choy, Michelle Cheong, Ma Nang Laik and Koo Ping Shung. US Presidential Election 2012 Prediction using Census Corrected Twitter Model. *Arxiv,* 2012.

[6] Matthew Conlen and Reuben Fischer-Baum. The Facebook Primary. http://projects.fivethirtyeight.com/facebook-primary/. April 2016.

[7] James R. A. Davenport. 2016 Election Issues, by the Numbers. http://www.ifweassume.com/2015/12/2016-election-issues-by-numbers.html. 2016.

[8] Edison Research. The Infinite Dial 2016. Edison Research and Tritron Digital. 2016.

[9] Alec Go, Richa Bhayani and Lei Huang. Twitter Sentiment Classification using Distant Supervision. Stanford Digital Library, 2009.

[10] Wu He, Shenghua Zha and Ling Li. Social media competitive analysis and text mining: A case study in the pizza industry. *International Journal of Information Management,* 33(3):464-472, June 2013.

[11] Hsiu-Fang Hsieh and Sarah E. Shannon. Three Approaches to Qualitative Content Analysis. *Sage Journals,* 2005.

[12] Virginia Hughes. How Forensic Linguistics Outed J.K. Rowling (Not to Mention James Madison, Barack Obama, and the Rest of Us). http://phenomena.nationalgeographic.com/2013/07/19/how-forensic-linguistics-outed-j-k-rowling-not-to-mention-james-madison-barack-obama-and-the-rest-of-us/. 2013.

[13] Shamanth Kumar, Geoffrey Barbier, Mohammad Ali Abbasi and Huan Liu. TweetTracker: An Analysis Tool for Humanitarian and Disaster Relief. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media.* 2011.

[14] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[15] Maryland State Board of Elections. 2016 Presidential Primary Election State Candidates List. http://www.elections.state.md.us/elections/2016/primary_candidates/gen_cand_lists_2016_3_001-.html. 2016.

[16] Walaa Medhat, Ahmed Hassan and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal,* 2014.

[17] Gilad Mishne. Experiments with Mood Classification in Blog Posts. In *1st Workshop on Stylistic Analysis Of Text For Information*, 2005.

[18] Alexander Pak and Patrick Paroubek. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In *Proceeding of LREC*, Valletta, Malta, 2010.

[19] Bo Pang, Lillian Lee and Shivakumar Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2002.

[20] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval,* 2(1):1-135, 2008.

[21] Sarah Perez. Analysis of social media did a better job at predicting Trump's win than the polls. https://techcrunch.com/2016/11/10/social-media-did-a-better-job-at-predicting-trumps-win-than-the-polls/. 2016.

[22] Pew Research Center. Election 2016: Campaigns as a Direct Source of News. http://www.journalism.org/2016/07/18/presidential-candidates-changing-relationship-with-the-web/. 2016.

[23] Scientific American. Twitter Language Reveals Political Belief. *Scientific American Minds,* 27(3):8, 2016.

[24] Fredrik Sommar and Milosz Wielondek. *Combining Lexicon- and Learning-based Approches for Imporved Peformance and Convenience in Sentiment Classification*. KTH Royal Institute of Technology, 2015.

[25] Karolina Sylwester and Matthew Purver. Twitter Language Use Reflects Psychological Differences between Democrats and Republicans. *PLOS ONE,* 2015.

Appendix

All programs developed for this thesis are included on the enclosed CD. The following is a list of files stored on the CD:

1) FEATpy: Python package containing software for the FEAT Extractor and FEAT Analyzer.
    a) extractor.py – Python script that extracts page information such as the person's id, name, and news feed, and stores the information into the MySQL database.
    b) pfExtractor.py – Script that extracts the number of followers each day and stores the information into the page_fans table in the MySQL database.
    c) sentiment.py – Script that contacts the sentiment analysis API that analyzes the provided text for its sentiment value.
    d) read.py – Script that determines the grade levels of provided text using each of the readability formulas and stores those levels into the database.
2) nltk_data: Contains NLTK data such as the corpus and stop words files.
3) Readability: Library that contains the Python script necessary for calculating the readability score.
4) FEATvisualizer: Source code for the website portion of the thesis. This includes many important aspects of the Django framework.
    a) home1.html – The HTML file for the homepage of the website
    b) view.py – File that contains the views for the website.
    c) models.py – Script that contains the models that reflect the data stored in the database. Each model maps to a single database table.

d) settings.py – Settings file for the Django Framework

e) urls.py – File that maps URLs to views

f) sandbox.py – Script that runs the Word Cloud generator to produce word cloud images.

5) static: JavaScript and CSS files for the website

a) chartController.js – JavaScript file that contains the makeDynamicChart2 function that makes the graphs on the website by interacting with Chart.js.

b) wordcloud.js – JavaScript file that creates word cloud images.

6) Images: Contains the images produced from the word cloud generator and includes the donkey and elephant images used to supply the shapes for the word clouds.

Vita

Haihoua Yang was born in 1991 in Oroville, California. After five years of living in California, she moved to Albemarle, North Carolina where she lived for the next three years. From there, she moved to Kannapolis, North Carolina where she finished middle school and high school. After high school, Haihoua entered Appalachian State University where she spent about one year considering other majors such as Anthropology and graphic arts. She then became interested in Computer Science and decided to pursue a Bachelor of Science degree. After obtaining her bachelor's degree in December 2014, she continued for two more years to pursue her Master of Science degree in Computer Science. During this time, she worked as a teaching assistant in several courses: Introduction to Computer Applications, Computer Science I, Computer Science II, and Database. In addition, for four years, she participated in the NSF S-STEM program, which introduced her to numerous ideas and advancements in Computer Science and allowed her to work on interesting research projects with her peers.