Language Trend Analysis Across Twitter in the 2016 United States Presidential Election

Luke Taylor
Dennis Walsh
Thomas Flaherty

December 13, 2016

Abstract

Social media has increased the availability and immediateness of communication. Everyday language now becomes abundant data that is easier than ever to source, sort and analyze. Our objective is to explore this data and identify trends in the 2016 US Presidential election. Trends are defined as the most frequent words or topics discussed in Twitter messages over time or by area. Extensive cleaning of data is necessary so that searches for trends can occur. These trends in turn become data products. Insights into emerging trends can be seen by identifying keywords and their associated words. Trends can be visualized over time and geography, because Twitter data has the ability to attach geographic information to the Tweets. The methods used will be applicable to any topic of interest that someone wishes to explore in the future.

1 Introduction

What does an analysis of Twitter data tell us about the 2016 Presidential election? In a Presidential election many pieces of data are analyzed to determine strategy and behavior. Polls, voting history, television reports, and newspaper reports all come into play. In recent years, social media has become part of the mix. Analysis of Twitter activity can provide another window into what is happening in the election and why.

Twitter is a popular and important social media platform. Twitter had 320 million monthly active users worldwide in the last quarter of 2015. 65 million of those were in the United States, or 1 in 5 Americans. Usage is highest with adults under age 50, residents of cities, and upper-income Americans [**Desilver**].

Twitter reflects the attitudes only of certain subsets of the United States population. Pew Research has found that "the reaction on Twitter to major political events and policy decisions often differs considerably from general public opinion" [**Desilver**]. Twitter opinions

run more negative than the general public. Twitter users are younger and lean Democratic. Twitter is broader than most public opinion surveys because those under 18, as well as those outside the United States, can participate [**Mitchell**].

Social media does not necessarily mean conversations in the public square. Often it can mean two groups, in separate houses, talking among themselves. This type of twitter conversation has been called conversation within "Polarized Crowds, where opposed groups talk about the same topic but mostly just to other group members" [**Desilver**]. Thus it is appropriate to run separate searches on "Trump" and "Clinton" to look at what those polarized crowds are discussing. A search for "Obama" provides a contrast to the current candidates.

Research of tweets is not easily used to predict an election result, as the users of Twitter represent a biased sample of the population [**Gayo-Avello**]. But tweets can still reflect what issues and values are important to each side.

2 Data Collection

A representative set of sample data has been added to a GitHub project and is available via a web link visible in the source code for this paper.

The data set is a continuous JSON stream provided by the public Twitter sample API. This particular end point of the API provides a representative sample of data that a private individual may consume. This is contrasted by the private Twitter firehose stream which streams all of the Twitter statuses. By reserving the firehose API to a select few consumers with the knowledge and technology to utilize the firehose stream, Twitter limits any issues that would be caused by an average developer making a mistake while programming against the firehose API.

Although the sample API does not provide all of the tweets all of the time, it does provide a representative sample of data, as heavily tweeted events tend to generate representative samples [**Morstatter**]. While a city council campaign might not show up appropriately in the twitter sample stream, a US presidential contest should.

The JSON data provided by the public sample streaming API contains many fields and is considered structured data. Twitter's data objects are provided here:
https://dev.twitter.com/overview/api
The data used for this project comes from capturing the raw json output of sample stream.

Because of the large amount of data and the desire to analyze trends over a longer span of time than Twitter provides via its public API, it was decided to store the data in a SQL Server 2016 database. Data is parsed from JSON into a database table. The parsed data

does not contain all possible fields and is indexed on the date field of the tweet.

For the purposes of this document a sample of the parsed data is stored on GitHub and used for this analysis. To reproduce the data capture and transformation techniques required to get the data to this stage, refer to the Appendix for more details. Once in SQL Server, the parsed data is limited to only the rows desired for analysis. There are 50000 data points and 13 variables in this data set. The variables are MessageId, CreatedAt, UserId, PlaceId, PlaceType, PlaceName, PlaceFullName, PlaceCountryCode, PlaceCountry, PlaceBoundingBox, CoordinatesType, CoordinatesCoordinates, Text.

Below is a function defining a call to the SQL Database which retrieves the parsed data along with a command which assigns the result set to a local variable. The query can be modified to limit results to specific time frames or only to tweets that contain a textual value using the T-SQL syntax available in SQL Server 2016.

```
GetDataSet <- function (term = NA, count = 1000) {
  count <- format(count,scientific = FALSE)
  library(RODBC)
  dbhandle <- odbcConnect("MyODBCConnectionName",
                          uid="MyUserId",
                          pwd="MyPassword")
  queryParts = c("SELECT TOP (",count,") [MessageId]
                        ,[CreatedAt]
                        ,[UserId]
                        ,[PlaceId]
                        ,[PlaceType]
                        ,[PlaceName]
                        ,[PlaceFullName]
                        ,[PlaceCountryCode]
                        ,[PlaceCountry]
                        ,[PlaceBoundingBox]
                        ,[CoordinatesType]
                        ,[CoordinatesCoordinates]
                        ,[Text]
                    FROM [TwitterData].[dbo].[NewFlattenedTweets]")
  if (!is.na(term)) {queryParts <- c(queryParts,
                                     " WHERE [Text] LIKE '%",term,"%'")}
  query <- paste(queryParts, collapse='')
  res <- sqlQuery(dbhandle, query)
  odbcClose(dbhandle)
  return (res);
}
```

```
# Usage
dataSet <- GetDataSet("Trump",10000);
```

This study collected tweets for 23 days from September 25 to October 19, 2016. The Twitter feed was interrupted on October 19 and could not be re-established. The reason for the end of data collection was two-fold. The source data was nearing the capacity of the server's available storage and the Twitter authorization mechanism was updated. This data collection and analysis is still a worthwhile study on the use of Twitter data.

3 Data Preparation

The most salient value of this data is with the combination of the words of the tweet and the geographic information included with the tweets. In order to prepare the data for textual analysis which includes word frequency counts, word relationships, and word sentiment, it will take the form of a corpus. The corpus structure used for this analysis is part of the `tm` package.

Before the textual data can be used, it must be prepared by stripping out characters that interfere with English word analysis. The various transformations may or may not be desired based on the type of analysis being performed. In order to encapsulate the process, it is wrapped in a function named `TwitterToCorpus` and passed a character vector consisting of the tweet text.

The `TwitterToCorpus` function first removes any references to the original search term, if given. It then coverts the text to lower case. Next it removes any URLs. Then it removes any Twitter mentions. Numbers and stopwords are removed next, along with any punctuation. Finally, any non-alpha characters are replaced with a space and redundant whitespace is stripped from the data.

Now the corpus is ready to examine the frequency of words. An out of memory issue will arise when trying to transform a textttTermDocumentMatrix into a standard R matrix. To overcome this issue, the textttslam package is used. The problem stems from the fact that, in a very large data matrix where, conceptually, each word represents a row and each column represents a document (each of the tweets is considered a document) the value of that intersection is the frequency for that word in that particular document. In this case, the matrix is large and very sparse. The slam package allows the matrix to be "rolled up" by collapsing each column of data using a function such as sum without first converting the source data into a matrix.

Table 1: Top words by frequency

Term	Freq
debatenight	9534
debates	6383
debate	5110
hillary	4954
donald	4255

4 Data Challenges

There are many interesting things to be learned from the data. Although the source data from the Twitter stream was limited by the query to English tweets, many tweets contained language other than English, including languages that require a the extended UTF character set to display. This indicates that although all twitter accounts were set to use English as the default language, a decent percentage of tweets were non-English.

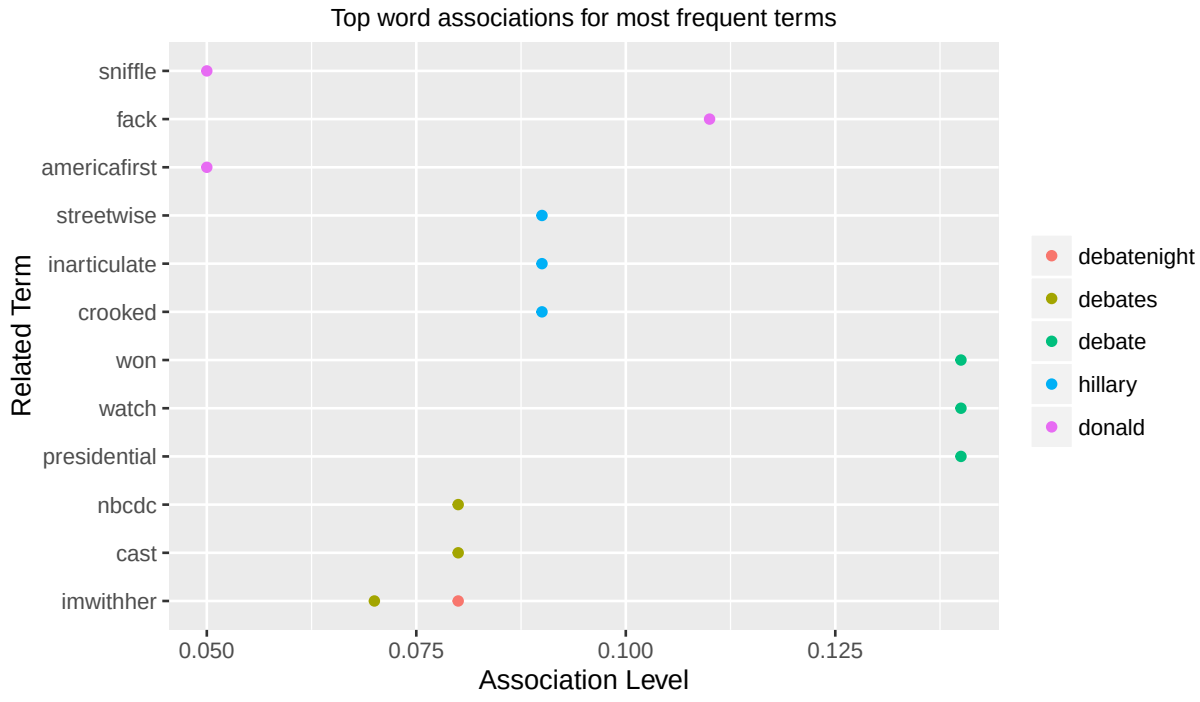
Tweets do not always provide a simple latitude, longitude to geolocate the tweet. Some contain city information, some contain a bounding box with surrounds an area where the tweet would have come from. This creates an addition layer of work in preparing the data and is due to various personal settings Twitter users adjust to control privacy levels.

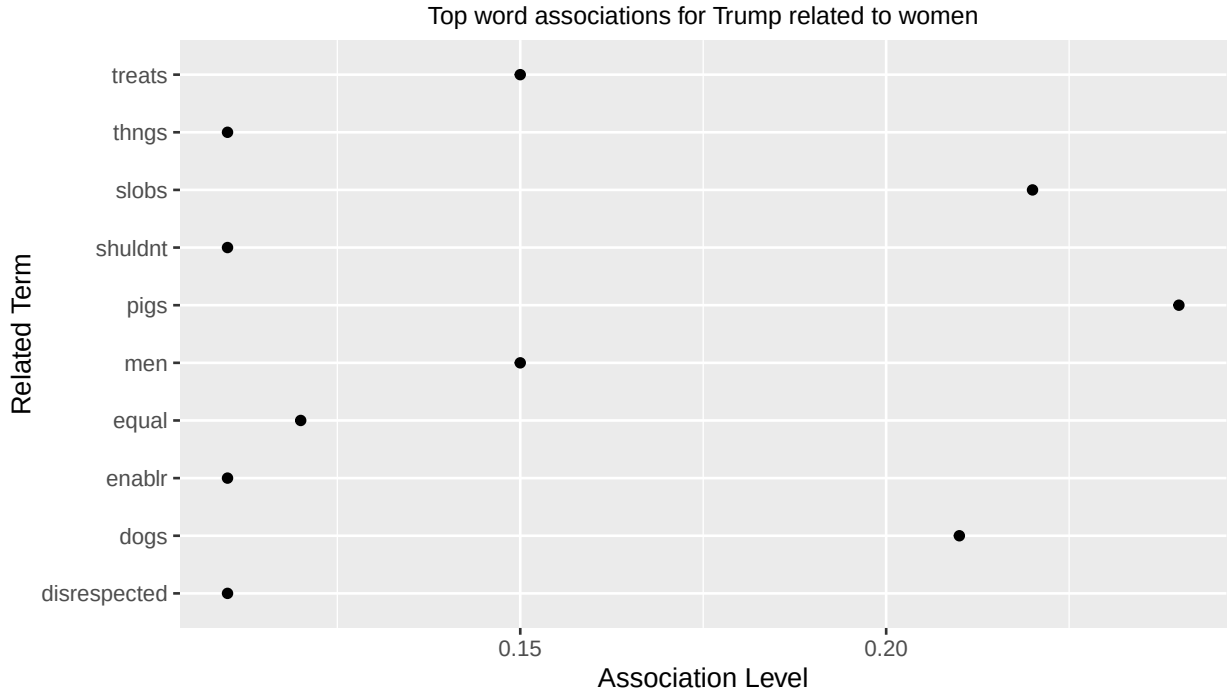
5 Data Product

Now that the word counting of the corpus is complete, and the word matrix has been rolled up into a frequency data frame, we can display the most frequent words in the data set. Collecting word frequencies is the first step in any analysis of twitter data. Table 1 presents the top words by frequency for tweets using the search term "Trump."

Next, we want to take a look at which words are most correlated with the top frequency words in the data set. Using the textttNLP and texttttm packages, we can get a measure of word associations. What this means is that we will find out how commonly a word appears in the same tweet within our data set. Some results in the Top Word Associations graph tend to be unimportant. Associated words such as "president", "watch", "tonights", and "presidential" do not lead the reader very far. But among other word associations, "crooked" shows that the phrase "crooked Hillary" is quite prominent in the social media discussion. The idea of polarized crowds is also apparent, as opinion leaders on each side are mentioned frequently. Word association "realalexjones" refers to conservative radio host Alex Jones,

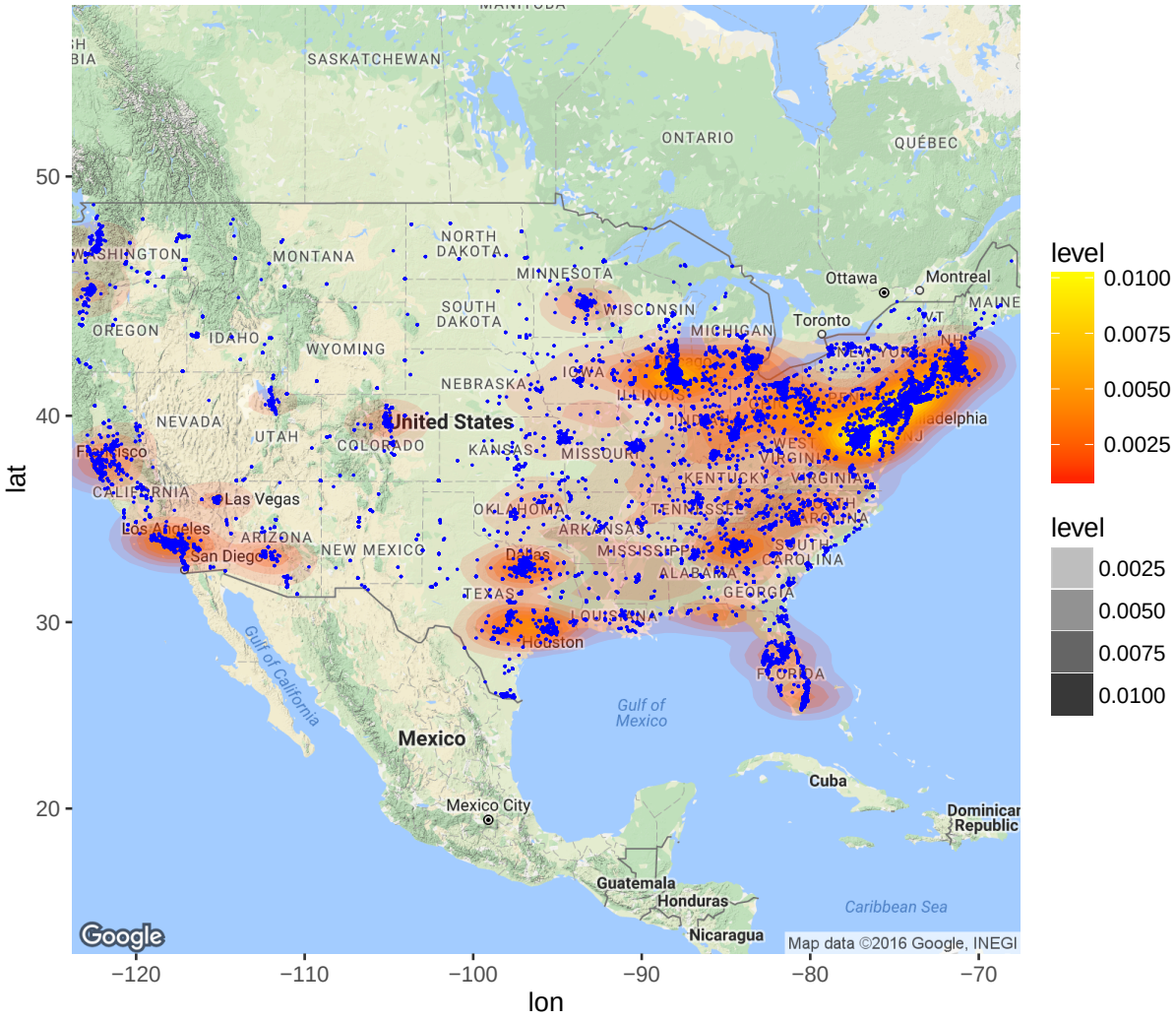
and "amjoy" to MSNBC host Joy Ann-Reid. Two separate reactions, rather than a discussion, are going on in social media.





Location information is extracted from each tweet, providing a heat map of tweet sources during the time period for the given search term. This data may provide insight on topic

interest by region. The tweet locations are mapped to show to show the origin of tweets relating to the given search term.



6 Conclusion

Donald Trump dominated in Twitter counts. The number of tweets mentioning Trump outnumbered Clinton by 3.2 to 1, and outnumbered President Obama by 11.8 to 1, over the 25 day range of the sample. Two dates stood out for exceptionally high traffic. September 27th was the day after the first presidential debate. October 10th was the day after the second

presidential debate. In both cases Trump mentions approximated 100,000 tweets, verssu 33,000 for Clinton, and 5,000 for President Obama.

date	Trump	Clinton	Obama
09/25/2016	113	38	26
09/26/2016	1812	735	123
09/27/2016	97267	32353	5168
09/28/2016	3119	1075	212
09/29/2016	0	0	0
09/30/2016	23140	7345	2474
10/01/2016	0	0	0
10/02/2016	18268	5756	1572
10/03/2016	20900	8278	2212
10/04/2016	21032	8809	3035
10/05/2016	12901	4028	769
10/06/2016	15991	6572	2684
10/07/2016	6452	2530	1110
10/08/2016	68775	15087	2778
10/09/2016	49886	13062	2389
10/10/2016	106596	33451	5154
10/11/2016	12846	3984	812
10/12/2016	29573	9891	2658
10/13/2016	43176	13395	5456
10/14/2016	38004	11329	6090
10/15/2016	33321	9832	4003
10/16/2016	30749	8826	2696
10/17/2016	27375	8369	2793
10/18/2016	31083	9181	4428
10/19/2016	2941	861	412

One consideration for future research is the rise of automated programs generating tweets in favor of one candidate - so called, "twitter bots." Samuel Woolley, the director of research at the Computational Propaganda project at Oxford University, found that one-third of all pro-Trump tweets, and one-fifth of pro-Clinton tweets, appeared to be automated [**Johnson**]. How can citizens learn to tell which are real opinions, and which are fake? Who wants to engage in conversation, and who does not?

The weaknesses of Twitter, and social media generally, were reflected in the 2016 election. The election played out as it did on Twitter: two polarized groups, talking among themselves. Twitter fails as an area of common ground. It reflects, but also helps to create, two separate groups, each with their own set of beliefs, and even their own facts. The future of common ground, community, and democracy depend upon finding some way to recreate the

ability, and even the desire, to have face to face conversations, rather than the often faceless interactions of Twitter.

7 Appendix

The SQL Server table schema

```
USE [TwitterData]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[NewFlattenedTweets](
[MessageId] [bigint] NOT NULL,
[CreatedAt] [datetime] NOT NULL,
[UserId] [bigint] NULL,
[PlaceId] [nvarchar](350) NULL,
[PlaceType] [nvarchar](350) NULL,
[PlaceName] [nvarchar](350) NULL,
[PlaceFullName] [nvarchar](350) NULL,
[PlaceCountryCode] [nvarchar](10) NULL,
[PlaceCountry] [nvarchar](350) NULL,
[PlaceBoundingBox] [nvarchar](350) NULL,
[CoordinatesType] [nvarchar](350) NULL,
[CoordinatesCoordinates] [nvarchar](350) NULL,
[Text] [nvarchar](max) NULL,
 CONSTRAINT [PK_NewFlattenedTweets] PRIMARY KEY CLUSTERED
(
[MessageId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
```

A sample of inserting data from twitter json source file into table.

```
create table #tempTweets(Json nvarchar(max))
```

```
BULK INSERT #tempTweets --RawTweetJson
FROM 'C:\Share\Tweets_20161013_clean.json'

select count(*) from #tempTweets
select top 10 * from #tempTweets

  --SELECT MAX(LEN(JSON_VALUE([Json], '$.text'))) [Text]
  --FROM #tempTweets TT


INSERT INTO NewFlattenedTweets
(     [MessageId]
    ,[CreatedAt]
      ,[UserId]--18
    ,[PlaceId]
      ,[PlaceType]
      ,[PlaceName]
      ,[PlaceFullName]
      ,[PlaceCountryCode]
      ,[PlaceCountry]
      ,[PlaceBoundingBox]
      ,[CoordinatesType]
      ,[CoordinatesCoordinates]
      ,[Text])--545
SELECT DISTINCT
       CAST(JSON_VALUE([Json], '$.id') AS BIGINT) MessageId
  --JSON_VALUE([Json], '$.created_at') CreatedAt
  ,CONVERT(DATETIME,SUBSTRING(JSON_VALUE([Json], '$.created_at'),4,7) +
SUBSTRING(JSON_VALUE([Json], '$.created_at'),26,5) +
SUBSTRING(JSON_VALUE([Json], '$.created_at'),11,9))
CreatedAt

  ,CAST(JSON_VALUE([Json], '$.user.id') AS BIGINT) UserId
  ,CAST(JSON_VALUE([Json], '$.place.id') AS NVARCHAR(350)) PlaceId
  ,CAST(JSON_VALUE([Json], '$.place.place_type') AS NVARCHAR(350)) PlaceType
  ,CAST(JSON_VALUE([Json], '$.place.name') AS NVARCHAR(350)) PlaceName
  ,CAST(JSON_VALUE([Json], '$.place.full_name') AS NVARCHAR(350)) PlaceFullName
  ,CAST(JSON_VALUE([Json], '$.place.country_code') AS NVARCHAR(10)) PlaceCountryCode
  ,CAST(JSON_VALUE([Json], '$.place.country') AS NVARCHAR(350)) PlaceCountry
  ,CAST(JSON_QUERY([Json], '$.place.bounding_box') AS NVARCHAR(350)) PlaceBoundingBox
  ,CAST(JSON_VALUE([Json], '$.coordinates.type') AS NVARCHAR(350)) CoordinatesType
  ,CAST(JSON_QUERY([Json], '$.coordinates.coordinates') AS NVARCHAR(350)) CoordinatesCoo
  ,CAST(JSON_VALUE([Json], '$.text') AS NVARCHAR(140)) [Text]
  FROM #tempTweets TT
```

```
  WHERE JSON_VALUE([Json], '$.created_at') IS NOT NULL
  AND CAST(JSON_VALUE([Json], '$.id') AS BIGINT) NOT IN (
  SELECT MessageId
  FROM [TwitterData].[dbo].[NewFlattenedTweets]
  )

drop table #tempTweets
```

References

[1] Desilver, Drew. "5 facts about Twitter at age 10". *Pew Research*, urlhttp://www.pewresearch.org/fact-tank/2016/03/18/5-facts-about-twitter-at-age-10/ . 2016

[2] Gayo-Avello, Daniel. "Don't Turn Social Media into Another Literary Digest Poll." *Communications of The ACM* 54.10 (2011): 121-128. *Business Source Elite*.

[3] Giachanou, Anastasia, and Fabio Crestani. "Like It or Not: A Survey of Twitter Sentiment Analysis Methods." *ACM Computing Surveys* 49.2 (2016): 28-28:41. *Business Source Elite*.

[4] Johnsonk, Tim. "That Pro-Trump Tweet that Made Your Blood Boil? It probably came from a bot." McClatchey News Service, November 4, 2016. urlhttp://www.mcclatchydc.com/news/politics-government/election/article112585668.html

[5] Kodali, Teja. "Building Wordclouds in R". 2015. urlhttps://www.r-bloggers.com/building-wordclouds-in-r/

[6] Mitchell, Amy and Paul Hitlin. "Events Often at Odds with Overall Public Opinion". *Pew Research*. 2013. urlHTTP://WWW.PEWRESEARCH.ORG/2013/03/04/TWITTER-REACTION-TO-EVENTS-OFTEN-AT-ODDS-WITH-OVERALL-PUBLIC-OPINION/

[7] Morstatter, Fred, Jrgen Pfeffer, Huan Liu, I& Kathleen M. Carley. "Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose". arXiv:1306.5204 [cs.SI]. 2013. urlhttps://arxiv.org/abs/1306.5204