Advanced Machine Learning - Final Project

Steve Spence, Charity Elijah, Elham Zare, Timothy Akintoye

5/1/2020

Twitter API has a limit of 200 tweets; therefore, the team had to utilize the following website to pull Trump historic tweets from:

<http://www.trumptwitterarchive.com>

This site allowed the team to pull tweets from a specific time frame. For the scope of this project, the team pulled the Twitter data around the time frame of COVID-19 (January 1, 2020 to Present)

# Import Trump's historic tweets  
  
require(readxl)

## Loading required package: readxl

Trump\_Tweets\_Test <- read\_excel("Trump\_Tweets\_2020.xlsx")  
  
head(Trump\_Tweets\_Test)

## # A tibble: 6 x 8  
## source text created\_at retweet\_count favorite\_count is\_retweet  
## <chr> <chr> <dttm> <dbl> <dbl> <lgl>   
## 1 Twitt… "RT … 2020-04-30 20:26:30 8420 0 TRUE   
## 2 Twitt… "RT … 2020-04-30 19:58:53 8421 0 TRUE   
## 3 Twitt… "Ove… 2020-04-30 18:25:29 16688 63862 FALSE   
## 4 Twitt… "RT … 2020-04-30 16:32:58 8583 0 TRUE   
## 5 Twitt… "RT … 2020-04-30 16:32:52 6159 0 TRUE   
## 6 Twitt… "RT … 2020-04-30 14:37:07 18099 0 TRUE   
## # … with 2 more variables: id\_str <dbl>, week\_no <dbl>

Next, we will only select the date and text columns.

require(tidytext)

## Loading required package: tidytext

require(tidyr)

## Loading required package: tidyr

require(dplyr)

## Loading required package: dplyr

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

require(rtweet)

## Loading required package: rtweet

tweets.Trump <- Trump\_Tweets\_Test %>% select(created\_at, text)

Now, we will have to clean up the tweets by: 1. Converting to lowercase 2. Revert words to stem words 3. Removing “[https://](NULL)” links 4. Removing punctuation 5. Removing stop words

# Remove hyperlink elements  
  
tweets.Trump$stripped\_text <- gsub("http\\S+","",tweets.Trump$text)  
  
# Convert words to lowercase, remove punctutation, and create an id for each tweet  
  
tweets.Trump.stem <- tweets.Trump %>%  
 select(stripped\_text) %>%  
 unnest\_tokens(word, stripped\_text)  
  
# Remove stop words from the output  
  
cleaned.tweets.Trump <- tweets.Trump.stem %>%  
 anti\_join(stop\_words)

## Joining, by = "word"

# Review the results  
  
head(cleaned.tweets.Trump)

## # A tibble: 6 x 1  
## word   
## <chr>   
## 1 rt   
## 2 whitehouse  
## 3 live   
## 4 potus   
## 5 delivers   
## 6 remarks

We can now look at the most popular words during this time frame

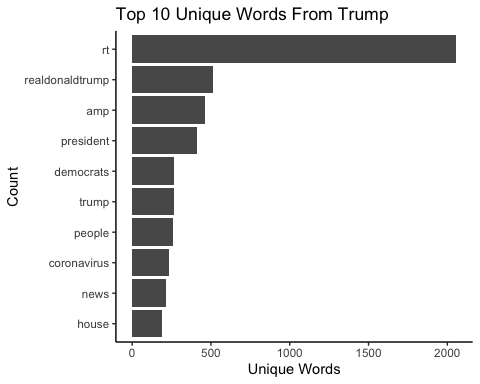
require(ggplot2)

## Loading required package: ggplot2

# Reveal the top 10 words during this timeframe  
  
top\_words <- cleaned.tweets.Trump %>%  
 count(word, sort = TRUE) %>%  
 top\_n(10) %>%  
 mutate(word = reorder(word, n)) %>%  
 ggplot(aes(x = word, y = n)) +  
 geom\_col() +  
 xlab(NULL) +  
 coord\_flip() +  
 theme\_classic() +  
 labs(x = "Count",  
 y = "Unique Words",  
 title = "Top 10 Unique Words From Trump")

## Selecting by n

print(top\_words)



Next, a sentitment analysis will be performed on the tweets.

Below shows example words that are considered “positive” ( values greater than 0 ) and “negative” ( values less than 0 ). Afinn will be used since it takes a score of the total words in the Tweet.

require(tidytext)  
require(textdata)

## Loading required package: textdata

# Examples of postiive words  
  
get\_sentiments("afinn") %>%  
 filter(value == "3")

## # A tibble: 172 x 2  
## word value  
## <chr> <dbl>  
## 1 admire 3  
## 2 admired 3  
## 3 admires 3  
## 4 admiring 3  
## 5 adorable 3  
## 6 adore 3  
## 7 adored 3  
## 8 adores 3  
## 9 affection 3  
## 10 affectionate 3  
## # … with 162 more rows

# Examples of negative words  
  
get\_sentiments("afinn") %>%  
 filter(value == "-3")

## # A tibble: 264 x 2  
## word value  
## <chr> <dbl>  
## 1 abhor -3  
## 2 abhorred -3  
## 3 abhorrent -3  
## 4 abhors -3  
## 5 abuse -3  
## 6 abused -3  
## 7 abuses -3  
## 8 abusive -3  
## 9 acrimonious -3  
## 10 agonise -3  
## # … with 254 more rows

Next, we will perform the sentitment analysis on the summation of all tweets with the “Afinn” lexicon.

# Sentiment analysis with "Afinn" lexicon.  
  
afinn.tweets.Trump <- cleaned.tweets.Trump %>%  
 inner\_join(get\_sentiments("afinn")) %>%  
 count(word, value, sort = TRUE) %>%  
 ungroup()

## Joining, by = "word"

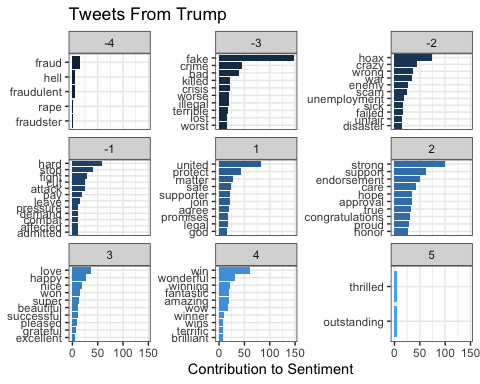
afinn.tweets.Trump

## # A tibble: 809 x 3  
## word value n  
## <chr> <dbl> <int>  
## 1 fake -3 147  
## 2 strong 2 100  
## 3 united 1 83  
## 4 hoax -2 75  
## 5 support 2 62  
## 6 win 4 62  
## 7 hard -1 58  
## 8 endorsement 2 51  
## 9 crime -3 46  
## 10 crazy -2 44  
## # … with 799 more rows

This chart will show a summary of all words Tweets during the desired timeframe and plot out the frequency of each word used.

# Summary count of all words tweeted  
  
afinn.tweets.Trump %>%  
 group\_by(value) %>%  
 top\_n(10) %>%  
 ungroup() %>%  
 mutate(word = reorder(word, n)) %>%  
 ggplot(aes(word, n, fill = value)) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~value, scales = "free\_y") +  
 labs(title = "Tweets From Trump",  
 y = "Contribution to Sentiment",  
 x = NULL) +  
 coord\_flip() +  
 theme\_bw()

## Selecting by n



However, we are more considered about the sentiment of each Tweet itself. Therefore, we will need to get a total score for each Tweet. The code for this is shown below.

# Sentiment Score for Each Tweet  
  
sentiment.afinn <- function(twt){  
 twt\_tbl = tibble(text = twt) %>%  
 mutate(  
 stripped\_text = gsub("http\\S+","", text)  
 ) %>%  
 unnest\_tokens(word, stripped\_text) %>%  
 anti\_join(stop\_words) %>%  
 inner\_join(get\_sentiments("afinn")) %>%  
 count(word, value, sort = TRUE) %>%  
 ungroup() %>%  
 mutate(  
 score = value  
 )  
   
 # Calculate total score for each tweet  
 sent.score = case\_when(  
 nrow(twt\_tbl) == 0 ~ 0,  
 nrow(twt\_tbl) > 0 ~ sum(twt\_tbl$score)  
 )  
   
 # Keep track of tweets that contain no words from afinn list  
   
 zero.type = case\_when(  
 # Type 1 Means No Words at all  
 nrow(twt\_tbl) == 0 ~ "Type 1",  
 # Type 2 Means Sum of All Words = 0  
 nrow(twt\_tbl) > 0 ~ "Type 2"  
 )  
   
 list(score = sent.score, type = zero.type, twt\_tbl = twt\_tbl)  
}

Now we will apply the function to the Tweets

# Apply the function to the set of tweets  
  
Trump.tweets.sent <- lapply(Trump\_Tweets\_Test$text, function(x){sentiment.afinn(x)})

require(dplyr)  
require(purrr)

## Loading required package: purrr

##   
## Attaching package: 'purrr'

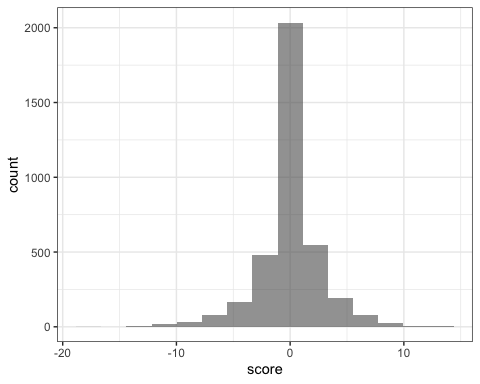
## The following object is masked from 'package:rtweet':  
##   
## flatten

Trump\_sentiment <- bind\_rows(  
 tibble(  
 date = Trump\_Tweets\_Test$created\_at,  
 score = unlist(map(Trump.tweets.sent, "score")),  
 type = unlist(map(Trump.tweets.sent, "type")),  
 tweet = Trump\_Tweets\_Test$text  
 )  
)  
  
Trump\_sentiment

## # A tibble: 3,673 x 4  
## date score type tweet   
## <dttm> <dbl> <chr> <chr>   
## 1 2020-04-30 20:26:30 0 Type 1 "RT @WhiteHouse: LIVE: POTUS Delivers Remar…  
## 2 2020-04-30 19:58:53 2 Type 2 "RT @StevenTDennis: Trump gets bump in late…  
## 3 2020-04-30 18:25:29 0 Type 1 "Over 120 MILLION Economic Impact Payments …  
## 4 2020-04-30 16:32:58 0 Type 1 "RT @WhiteHouse: President @realDonaldTrump…  
## 5 2020-04-30 16:32:52 -1 Type 2 "RT @WhiteHouse: President @realDonaldTrump…  
## 6 2020-04-30 14:37:07 -3 Type 2 "RT @JudiciaryGOP: We already knew that Jam…  
## 7 2020-04-30 14:09:10 -3 Type 2 "We can’t let the Fake News and their partn…  
## 8 2020-04-30 13:17:50 0 Type 1 "RT @RepStefanik: Just announced: over $33 …  
## 9 2020-04-30 13:16:56 0 Type 1 "RT @RepLizCheney: Speaker Pelosi still ref…  
## 10 2020-04-30 13:16:43 2 Type 2 "RT @RepLizCheney: Safeguarding our nation’…  
## # … with 3,663 more rows

Now we can plot out a histogram of the sentiments for review.

require(ggplot2)  
  
# Plot of the tweet sentitments  
  
ggplot(Trump\_sentiment, aes(x = score)) +  
 geom\_histogram(bins = 15, alpha = 0.6) +  
 theme\_bw()



We will also export the result as a CSV, so we can attempt to plot out the results in another software program.

# Return a CSV of the file  
  
write.csv(Trump\_sentiment,"sentitments.csv",row.names = TRUE)

The exported CSV files were then analyzed in Excel to create visually appealing graphs for our story.