Data607_HW10_Connin

04/17/21

Project Overview

This project is an initial comparative sentiment analysis of 2016 presidential campaign speeches given by Hillary Clinton and Donald Trump. It addresses several questions:

1. How do the candidates compare in terms of the number and tone of their speeches?
2. Are there identifiable trends in the tone and message of each presidential candidate during the course of the election season?

Methods:

Campaign speeches collected for this analysis were obtained from:

Brown, D. W. (2017) Clinton-Trump Corpus. Retrieved from http://www.thegrammarlab.com (http://www.thegrammarlab.com).

The Corpus is described as, “a collection of speeches delivered at campaign events by Hillary Clinton and Donald Trump, beginning with their acceptance speeches at their respective party conventions and continuing up to the election.The corpus contains approximately from 114,000 words from Clinton and 440,000 words from Trump.”

This NLP study was conducted using the following text mining packages in R:

1. tm
2. tidytext
3. text data

Two lexicons (bing, nrc) were used to generate sentiment scores for each speech text. In turn each speech was tokenized via. individual words. Scores were compared for each candidate and speech across the campaign season. In addition, cumulative word counts (positive vs. negative) were compared between the candidates.

The raw data and R code for this analysis are available on Github at: https://github.com/sconnin/DATA607_HW10 (https://github.com/sconnin/DATA607_HW10).

```
library(tm)
library(stringr)
library(tidytext)
library(textdata)
library(tidyverse)
library(magrittr)
library(forcats)
library(cowplot)
library(kableExtra)
```

1. Create a corpus for each candidate.

The corpus includes all speeches given by each candidate during the 2016 campaign season

```
#Create corpus for Trump and Clinton

wdt<- "C:/Users/seanc/Documents/Data_Science/CUNY/Data 607 Acquisition and Management/Clinton-Trump Corpus/Trump"

wdc<- "C:/Users/seanc/Documents/Data_Science/CUNY/Data 607 Acquisition and Management/Clinton-Trump Corpus/Clinton"

vct <- VCorpus(DirSource(wdt))
vcc <- VCorpus(DirSource(wdc))
```

2. Clean each corpus.

```
get_stopwords()

#Clean Trump corpus

vct <- tm_map(vct, content_transformer(function(vct) gsub(vct, pattern ='<.*?>', replacement = "")))
vct <- tm_map(vct, removePunctuation)
vct <- tm_map(vct, content_transformer(tolower)) #Transform to lower case
vct <- tm_map(vct, removeNumbers) #Strip digits
vct <- tm_map(vct, stripWhitespace) #Strip whitespace (cosmetic?)

#Clean Clinton Corpus

vcc <- tm_map(vcc, content_transformer(function(vcc) gsub(vcc, pattern ='<.*?>', replacement = "")))
vcc <- tm_map(vcc, removePunctuation)
vcc <- tm_map(vcc, content_transformer(tolower)) #Transform to lower case
vcc <- tm_map(vcc, removeNumbers) #Strip digits
vcc <- tm_map(vcc, stripWhitespace) #Strip whitespace (cosmetic?)
```

3. Remove stopwords and tidy each corpus

```
#Tidy Trump corpus and remove stopwords

vcorpt <- vct %>% tidy()

vcorpt %<>% select(id, text)

vcorpt %<>% unnest_tokens(word, text)

vcorpt %<>% select(id, word)

vcorpt %<>% anti_join(get_stopwords())

#Tidy Clinton corpus and remove stopwords

vcorpc <- vcc %>% tidy()

vcorpc %<>% select(id, text)

vcorpc %<>% unnest_tokens(word, text)

vcorpc %<>% select(id, word)

vcorpc %<>% anti_join(get_stopwords())

# inspect Clinton corpus as example

head(vcorpc, 5)
```

```
## # A tibble: 5 x 2
##   id                    word
##   <chr>                 <chr>
## 1 Clinton_2016-07-28.txt thank
## 2 Clinton_2016-07-28.txt thank
## 3 Clinton_2016-07-28.txt much
## 4 Clinton_2016-07-28.txt thank
## 5 Clinton_2016-07-28.txt thank
```

4. Evaluate bing and nrc lexicons

Note: due to time other lexicons (e.g., affin, syuzhet) were not included in this project.

```
# Review lexicon prior to analysis in order to assess analysis steps

get_sentiments( "nrc") # assigns category of emotion to word
```

```
## # A tibble: 13,901 x 2
##    word        sentiment
##    <chr>       <chr>
##  1 abacus      trust
##  2 abandon     fear
##  3 abandon     negative
##  4 abandon     sadness
##  5 abandoned   anger
##  6 abandoned   fear
##  7 abandoned   negative
##  8 abandoned   sadness
##  9 abandonment anger
## 10 abandonment fear
## # ... with 13,891 more rows
```

```
get_sentiments( "bing") #assigns categories of positive or negative to word
```

```
## # A tibble: 6,786 x 2
##    word        sentiment
##    <chr>       <chr>
##  1 2-faces     negative
##  2 abnormal    negative
##  3 abolish     negative
##  4 abominable  negative
##  5 abominably  negative
##  6 abominate   negative
##  7 abomination negative
##  8 abort       negative
##  9 aborted     negative
## 10 aborts      negative
## # ... with 6,776 more rows
```

5. Convert each corpus to dataframe and subset using sentiment lexicons.

```
#Create dataframe for Trump sentiment scores using bing and nrc lexicons

nrc_bing_t <- data.frame()

lex<-c("nrc","bing")

for (i in 1:length(lex)){
    print(lex[i])
    df <- vcorpt %>%
    inner_join(get_sentiments(lex[i])) %>%
    count(id, sentiment) %>%
    spread(sentiment, n, fill = 0)%>%
    mutate(sentiment = positive - negative)%>%
    mutate(method = lex[i])%>%
    select(id, method, positive, negative, sentiment)
    nrc_bing_t <- rbind(df,nrc_bing_t)%>%arrange(id, method)
}
```

```
## [1] "nrc"
```

```
## Joining, by = "word"
```

```
## [1] "bing"
```

```
## Joining, by = "word"
```

```
#Create dataframe for Clinton sentiment scores using bing and nrc lexicons

nrc_bing_c <- data.frame()

for (i in 1:length(lex)){
    print(lex[i])
    df <- vcorpc %>%
    inner_join(get_sentiments(lex[i])) %>%
    count(id, sentiment) %>%
    spread(sentiment, n, fill = 0)%>%
    mutate(sentiment = positive - negative)%>%
    mutate(method = lex[i])%>%
    select(id, method, positive, negative, sentiment)
    nrc_bing_c <- rbind(df,nrc_bing_c)%>%arrange(id, method)
}
```

```
## [1] "nrc"
```

```
## Joining, by = "word"
```

```
## [1] "bing"
```

```
## Joining, by = "word"
```

```
# Print Clinton dataframe as example

head(nrc_bing_c, 5)
```

```
## # A tibble: 5 x 5
##   id                     method positive negative sentiment
##   <chr>                  <chr>     <dbl>    <dbl>     <dbl>
## 1 Clinton_2016-07-28.txt bing        284      116       168
## 2 Clinton_2016-07-28.txt nrc         369      157       212
## 3 Clinton_2016-07-29.txt bing         81       17        64
## 4 Clinton_2016-07-29.txt nrc          88       27        61
## 5 Clinton_2016-08-01.txt bing         49       33        16
```

6. Evaluate summary statistics for speeches by candidate - word counts and sentiment scores

Note: count and sentiment statistics were not normalized to account for length of speech.

```
# Evaluation summary statistics for sentiment analysis using bing and nrc

#Summary statistics for Trump campaign

nrc_bing_t%>%summary()
```

```
##       id              method            positive        negative
##  Length:164         Length:164         Min.   : 48.0   Min.   : 25.0
##  Class :character   Class :character   1st Qu.:171.0   1st Qu.:123.5
##  Mode  :character   Mode  :character   Median :228.5   Median :162.0
##                                        Mean   :231.1   Mean   :156.0
##                                        3rd Qu.:285.5   3rd Qu.:197.2
##                                        Max.   :510.0   Max.   :296.0
##    sentiment
##  Min.   :-104.00
##  1st Qu.:  34.50
##  Median :  63.50
##  Mean   :  75.08
##  3rd Qu.: 100.25
##  Max.   : 319.00
```

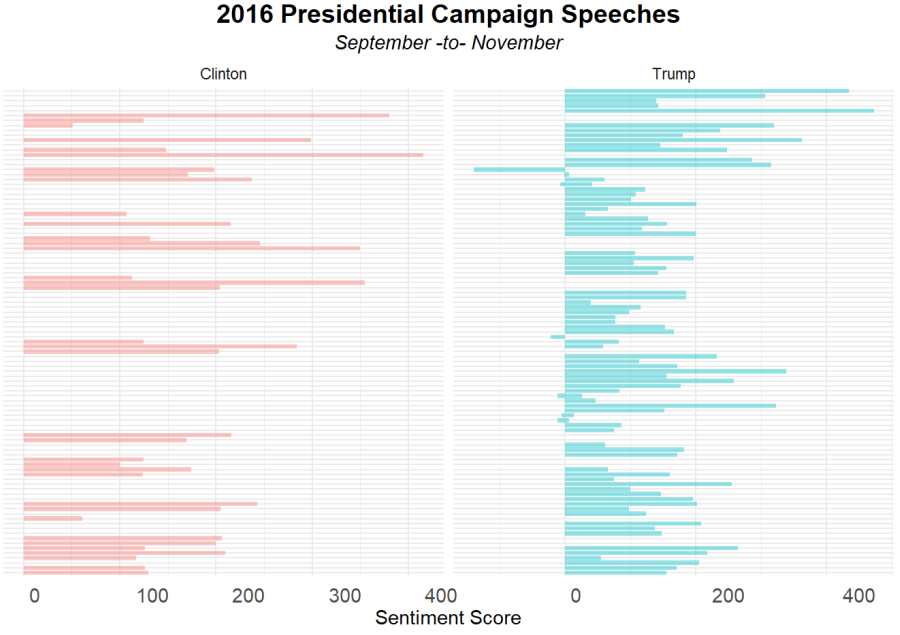
```
# Summary statistics for Clinton campaign

nrc_bing_c%>%summary()
```

```
##       id              method            positive       negative
##  Length:72          Length:72          Min.   : 37.0   Min.   :  5.00
##  Class :character   Class :character   1st Qu.:109.5   1st Qu.: 41.50
##  Mode  :character   Mode  :character   Median :155.5   Median : 62.00
##                                        Mean   :162.4   Mean   : 66.71
##                                        3rd Qu.:194.0   3rd Qu.: 91.75
##                                        Max.   :379.0   Max.   :157.00
##    sentiment
##  Min.   : 16.00
##  1st Qu.: 63.75
##  Median : 86.00
##  Mean   : 95.71
##  3rd Qu.:119.00
##  Max.   :233.00
```

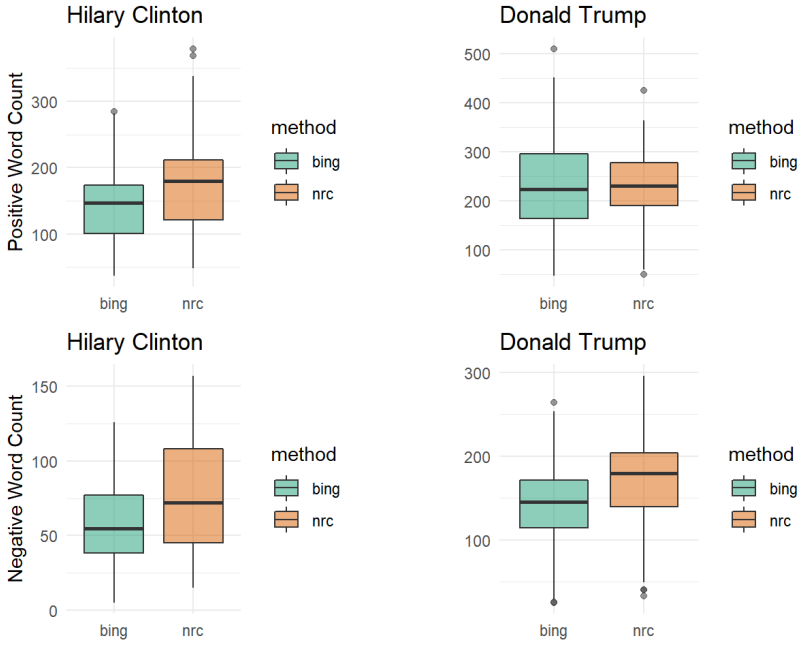
7. Compare sentiment scores and word counts.

Note: word count comparisons were limited to the 20 most frequent words for both positive and negative associations.

```
# Create dataframe for combined Clinton & Trump sentiment scores by date of campaign speech

nrc_bing_c%<>%mutate(Candidate = "Clinton")
nrc_bing_t%<>%mutate(Candidate = "Trump")

c_t <- rbind(nrc_bing_c, nrc_bing_t)%>%arrange(id, method)%>%
    mutate(id=gsub(".*_(.*)\\..*", "\\1", id))%>% #get just the day and month for a Date column
    separate(id, into = c("temp","Date"), sep="2016-" )%>%
    select(!temp)%>%
    rowid_to_column("id")
c_t$Date <- gsub( '-', '/', c_t$Date)

# Compare sentiment scores by each candidate over campaign season

ggplot(c_t, aes(Date, sentiment, fill = Candidate)) +
    geom_bar(alpha = 0.4, stat = "identity", show.legend = FALSE) +
    facet_wrap(~ Candidate, ncol = 2, scales = "free_x")+
    labs(y = "Sentiment Score", x = NULL)+
    theme_minimal()+
    theme(plot.title=element_text(size=14, hjust=0.5, face="bold", colour="black", vjust=-1))+
    theme(plot.subtitle=element_text(size=11, hjust=0.5, vjust=-0.5, face="italic", color="black"))+
    theme(axis.text.y=element_blank(),axis.ticks.y=element_blank())+
    theme(axis.text.x = element_text(size = 11, hjust = -0.5, vjust = .5))+
    ggtitle("2016 Presidential Campaign Speeches", subtitle = "September -to- November")+
    scale_x_discrete(limits=rev)+
    coord_flip()
```



```
# boxplot positive negative totals

hcp<-nrc_bing_c%>%ggplot(aes(x=method, y=positive, fill=method))+
  geom_boxplot(alpha=0.5)+
  ggtitle("Hilary Clinton")+
  labs(y = "Positive Word Count", x = NULL)+
  scale_fill_brewer(palette="Dark2")+
  theme(axis.text.y=element_text(size = 11))+
  theme_minimal()


hcn<-nrc_bing_c%>%ggplot(aes(x=method, y=negative, fill=method))+
    geom_boxplot(alpha=0.5)+
  ggtitle("Hilary Clinton")+
  labs(y = "Negative Word Count", x = NULL)+
  scale_fill_brewer(palette="Dark2")+
  theme(axis.text.y=element_text(size = 11))+
  theme_minimal()

dtp<-nrc_bing_t%>%ggplot(aes(x=method, y=positive, fill=method))+
    geom_boxplot(alpha=0.5)+
  ggtitle("Donald Trump")+
  labs(y = NULL, x = NULL)+
  scale_fill_brewer(palette="Dark2")+
  theme(axis.text.y=element_text(size = 11))+
  theme_minimal()

dtn<-nrc_bing_t%>%ggplot(aes(x=method, y=negative, fill=method))+
    geom_boxplot(alpha=0.5)+
  ggtitle("Donald Trump")+
  labs(y = NULL, x = NULL)+
  scale_fill_brewer(palette="Dark2")+
  theme(axis.text.y=element_text(size = 11))+
  theme_minimal()

# Convert plots to grob so that they can be grid plotted

as_grob(hcp)
as_grob(hcn)
as_grob(dtp)
as_grob(dtn)

ggdraw() +
  draw_plot(hcp, x = 0, y = .5, width = .4, height = .5) +
  draw_plot(dtp, x = .5, y = .5, width = .4, height = .5)+
  draw_plot(hcn, x = 0, y = 0, width = .4, height = .5) +
  draw_plot(dtn, x = .5, y = 0, width = .4, height = .5)
```



```
# Prepare corpus for evaluation of pos/neg word counts by candidate


wcounts_c <- vcorpc %>%
    as.data.frame()%>%
    group_by(word)%>%
    count(word) %>% arrange(desc(n, word))%>%
    inner_join(get_sentiments("bing"))%>%
    mutate(sentiment = ifelse(word == "trump" & sentiment == "positive", "negative", sentiment))
```

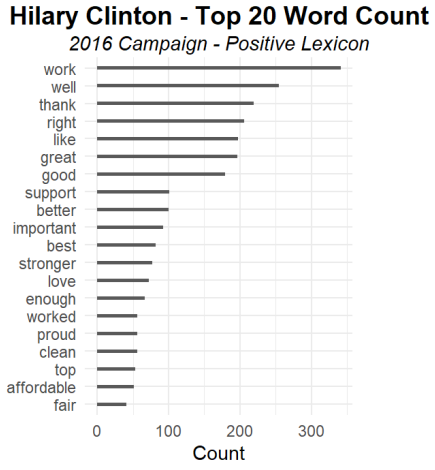
```
## Joining, by = "word"
```

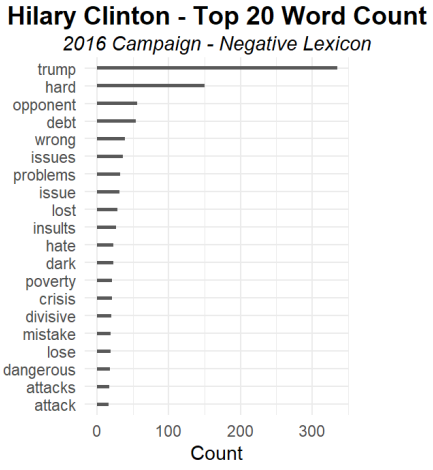
```
wcounts_t <- vcorpt %>%
    as.data.frame()%>%
    group_by(word)%>%
    count(word) %>% arrange(desc(n, word))%>%
    inner_join(get_sentiments("bing"))%>%
    mutate(sentiment = ifelse(word == "hilary" & sentiment == "positive", "negative", sentiment))
```

```
## Joining, by = "word"
```

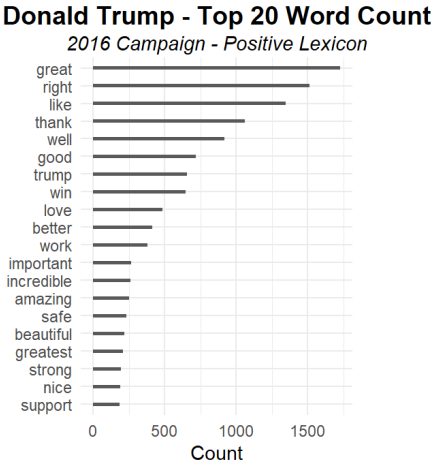
```
wcounts_c<-wcounts_c %>%
    mutate(word = fct_reorder(word, n))%>%
    pivot_wider(names_from=sentiment, values_from = n)

wcounts_t<-wcounts_t %>%
    mutate(word = fct_reorder(word, n))%>%
    pivot_wider(names_from=sentiment, values_from = n)


# Limit counts to top 20 pos & neg associations for both candidates. Separate dfs by association and candidate to enable gra
ph comparison.

c_pos<-wcounts_c%>%
    select(word, positive)%>%
    arrange(desc(positive))%>%
    head(20)

c_neg<-wcounts_c%>%
    select(word,negative)%>%
    arrange(desc(negative))%>%
    head(20)

t_pos<-wcounts_t%>%
    select(word, positive)%>%
    arrange(desc(positive))%>%
    head(20)

t_neg<-wcounts_t%>%
    select(word,negative)%>%
    arrange(desc(negative))%>%
    head(20)

# Build base plots for Clinton and Trump word counts

cp<-c_pos%>%
    ggplot(aes(reorder(word, positive), positive)) +
    geom_col(width = 0.2, show.legend = FALSE) +
    labs(y = "Count", x = NULL)+
    ggtitle("Hilary Clinton - Top 20 Word Count", subtitle = "2016 Campaign - Positive Lexicon")+
    theme_minimal()+
    theme(plot.title=element_text(size=14, hjust=0.5, face="bold", colour="black", vjust=-1))+
    theme(plot.subtitle=element_text(size=11, hjust=0.5, vjust=-0.5, face="italic", color="black"))+
    coord_flip()

cn<-c_neg%>%
    ggplot(aes(reorder(word, negative), negative)) +
    geom_col(width = 0.2, show.legend = FALSE) +
    labs(y = "Count", x = NULL)+
    ggtitle("Hilary Clinton - Top 20 Word Count", subtitle = "2016 Campaign - Negative Lexicon")+
    theme_minimal()+
    theme(plot.title=element_text(size=14, hjust=0.5, face="bold", colour="black", vjust=-1))+
    theme(plot.subtitle=element_text(size=11, hjust=0.5, vjust=-0.5, face="italic", color="black"))+
    coord_flip()

tp<-t_pos%>%
    ggplot(aes(reorder(word, positive), positive)) +
    geom_col(width = 0.2, show.legend = FALSE) +
    labs(y = "Count", x = NULL)+
    ggtitle("Donald Trump - Top 20 Word Count", subtitle = "2016 Campaign - Positive Lexicon")+
    theme_minimal()+
    theme(plot.title=element_text(size=14, hjust=0.5, face="bold", colour="black", vjust=-1))+
    theme(plot.subtitle=element_text(size=11, hjust=0.5, vjust=-0.5, face="italic", color="black"))+
    coord_flip()

tn<-t_neg%>%
    ggplot(aes(reorder(word, negative), negative)) +
    geom_col(width = 0.2, show.legend = FALSE) +
    labs(y = "Count", x = NULL)+
    ggtitle("Donald Trump - Top 20 Word Count", subtitle = "2016 Campaign - Negative Lexicon")+
    theme_minimal()+
    theme(plot.title=element_text(size=14, hjust=0.5, face="bold", colour="black", vjust=-1))+
    theme(plot.subtitle=element_text(size=11, hjust=0.5, vjust=-0.5, face="italic", color="black"))+
    coord_flip()

# Convert base plots to grob in order to grid plot - plot the col graphs

as_grob(cp)
as_grob(cn)
as_grob(tp)
```

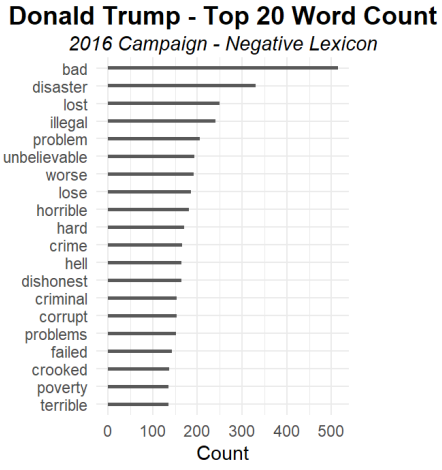
```
as_grob(tn)

ggdraw() +
  draw_plot(cp, x = 0, y = .2, width = .4, height = .75) +
  draw_plot(cn, x = .5, y = .2, width = .4, height = .75)
```






```
ggdraw() +
  draw_plot(tp, x = 0, y = .2, width = .4, height = .75) +
  draw_plot(tn, x = .5, y = .2, width = .4, height = .75)
```





Project Findings

Key observations:

1. The majority of speeches given by either candidate returned sentiment scores > 0, indicating a prevalence of positive word choices. Five speeches given by Donald Trump returned negative sentiment scores compared to zero for Clinton. Trump held four times the number of speeches than Clinton during the campaign season.
2. Comparison of sentiment scores for speeches given across the campaign season (September -to- November) reveals an overall decrease in positive messaging through time by both candidates.
3. The bing sentiment lexicon returned fewer word matches (positive & negative) relative to the nrc lexicon for both candidates.

4. Comparison of top 20 word counts (negative & positive) for each candidate indicates a higher proportion of pos:neg word counts for both candidates. Comparisons between the counts for each candidate should not be made until the count totals are normalized to account for the number of speeches each candidate gave.
5. The highest negative word count for Clinton was the term, “Trump”. Interestingly, references to Clinton do not appear in the top 20 word matches for speeches given by Trump. At first approximation, Trump’s word choices (positive & negative) appear more emotionally tinged than Clinton’s choices.

Recommendations:

The project findings are limited by the small ensemble of lexicon’s used to create sentiment word matches. The analysis could benefit from inclusion of additional lexicon’s. Similarly, custom word lists should be constructed to refine stop-word matches as well as sentiment matches based on further review of the speeches.

A more meaningful evaluation of the candidate’s motivations and use of speech might be accomplished through an analysis of word concordance and sentence-to-paragraph level tokenization. Other forms of NLP analysis can also be applied. This might include:

1. Topic Modelling
2. Word-Topic Probabilities
3. Document-topic Probabilities
4. Hierarchical Clustering
5. K-Means Clustering
6. Network Graphs

Updates to this project should also focus on standardizing the data to account for differences in the number and duration of speeches given by the candidates.

This project was inspired/informed by the following articles:

https://www.cell.com/patterns/pdf/S2666-3899(20)30005-2.pdf (https://www.cell.com/patterns/pdf/S2666-3899(20)30005-2.pdf)

https://uc-r.github.io/sentiment_analysis (https://uc-r.github.io/sentiment_analysis)