Untitled

ERICK@GURU

2024-03-29

Sentiment Analysis using R

sentiment Analysis of the US presidential debate 2016 It is a process of analyzing pieces of texts either from pdfs, webpages, social media posts, etc. to understand the kinds of emotions being expressed. Millions of texts are being sent everyday on tonnes of subjects, and it would be impossible to extract actionable insights from this data simply by reading each text, each Facebook post, tweet.

Sentiment Analysis provides easy way of extracting actionable insights through algorithms developed for programming Languages. For python check out some of the packages designed for sentiment analysis here. For R users some packages include;

- tm
- SentimentAnalysis
- tidytext
- quanteda

For this Analysis I will use the***tidytext***package because of its easy implementation alongside the tidyverse package. Find the data used hereand as well the *R script*

```
# Install
# install.packages("tm")  # for text mining
# install.packages("SnowballC") # for text stemming
# install.packages("wordcloud") # word-cloud generator
# install.packages("RColorBrewer") # color palettes
# install.packages("syuzhet") # for sentiment analysis
# install.packages("ggplot2") # for plotting graphs
# Load
rm(list=ls())#clean working directory
library(tidytext)
library("tm")
library("SnowballC")
library("wordcloud")
library("RColorBrewer")
library("syuzhet")
library("ggplot2")
library(reshape2)
library(tidyr)
```

```r
# ls("package:tidytext")
##load data
library(readr)
debate <- read_csv("C:/Users/langa/OneDrive/Desktop/Dataset/sentimentAnalysis_debate_data.csv")
names(debate)
```

```
## [1] "Line"    "Speaker" "Text"    "Date"
```

The first procedure is to examine the word frequencies by the speakers in the debate

```r
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 4.3.2
```

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.3
```

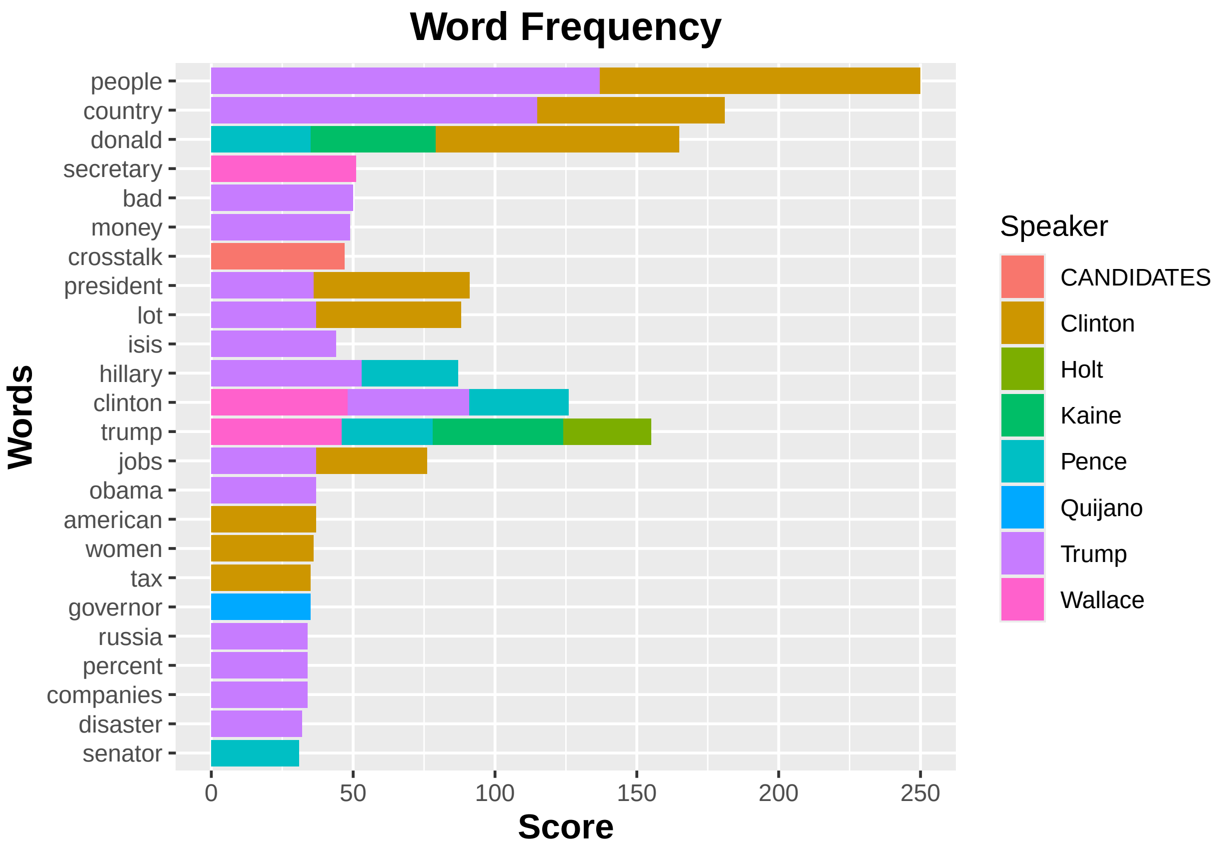
```
##
## Attaching package: 'dplyr'
```

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

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

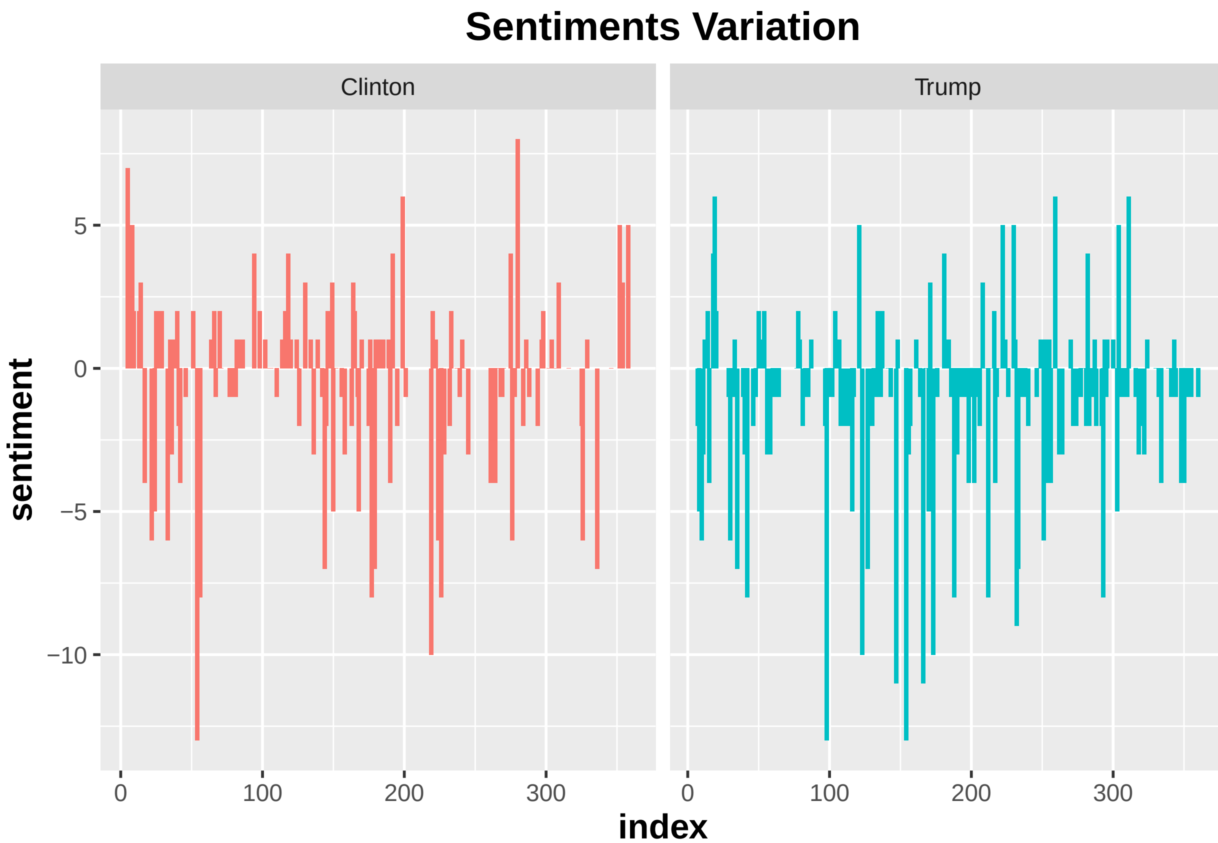
```r
str_extract <- stringr::str_extract
mutate <- dplyr::mutate
debate %>%
  group_by(Speaker) %>%
  unnest_tokens(word, Text) %>% #Tokenization
  #group_by(Speaker) %>%
  anti_join(stop_words) %>% #remove stop words
  count(word, sort = T) %>%
  mutate(word = str_extract(word, "[a-z]+")) %>%
  na.omit() %>%
  filter(n > 30) %>% #Extract words with frequencies > 30
  ggplot(., aes(reorder(word, n), n, fill = Speaker)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  ylab("Score") +
  xlab("Words") + ggtitle("Word Frequency") +
  theme(plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
        axis.title.x = element_text(face = "bold", size = 13),
        axis.title.y = element_text(face = "bold", size = 13))
```

```
## Joining with 'by = join_by(word)'
```



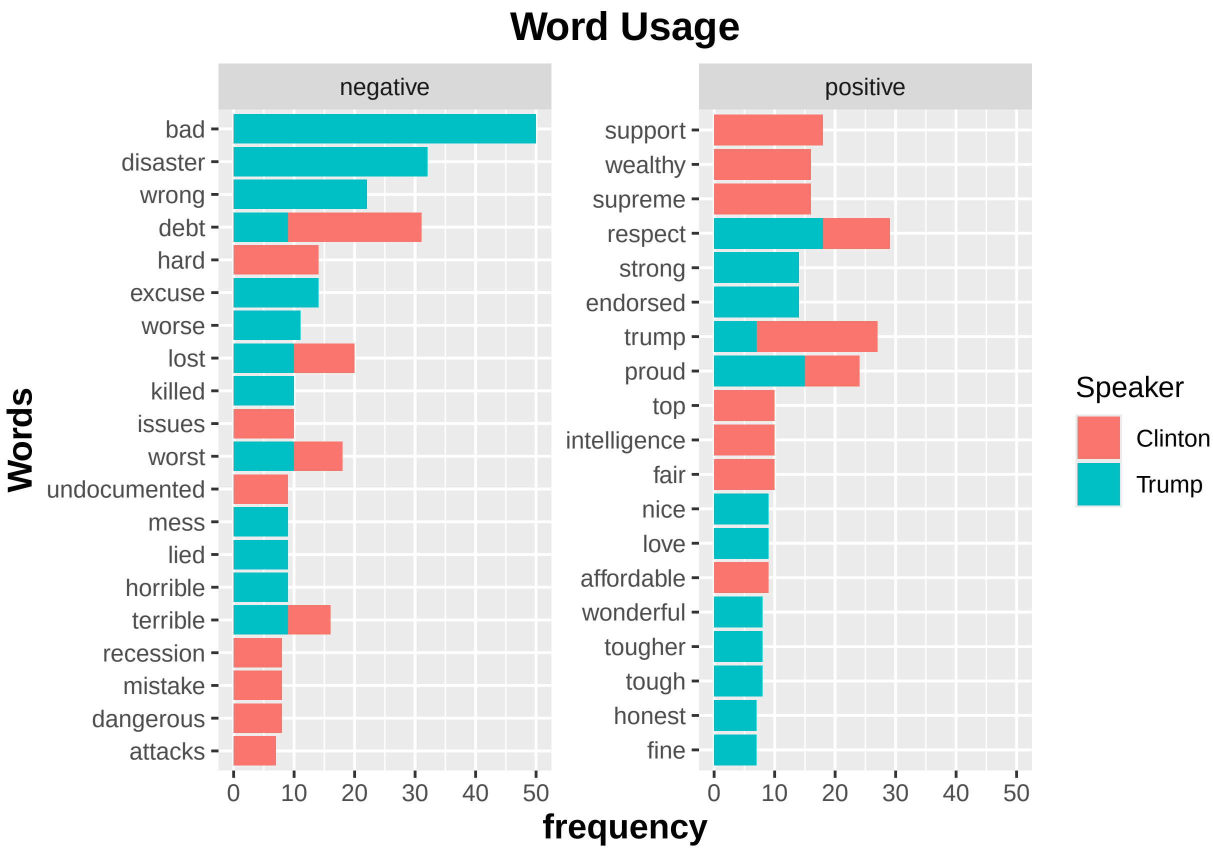
From the plot above, **Trump** and **Clinton** more than anyone else have more word frequencies, so lets place a little more interest in them. Lets examine how sentiments vary over the course of their speeches.

```
#Get the sentiments Variation
debate %>%
  filter(Speaker %in% c("Trump","Clinton")) %>%
  unnest_tokens(word, Text) %>%
  anti_join(stop_words) %>%
  inner_join(get_sentiments("bing")) %>%
  count(Speaker, index = Line, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative) %>%
  ggplot(.,aes(index, sentiment, fill = Speaker)) +
           geom_col(show.legend = FALSE, width = 3) +
           facet_wrap(~Speaker, ncol = 18, scales = "free_x") +
  ggtitle("Sentiments Variation") +
  theme(plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
        axis.title.x = element_text(face = "bold", size = 13),
        axis.title.y = element_text(face = "bold", size = 13))
```



It appears they both used quite a number of negative words in their speeches as compared to positive, so lets compare the speakers use of both words.

```
#plot a comparison of postive and negative words used by participant speakers (Trump vs Clinton)
debate %>%
  filter(Speaker %in% c("Trump","Clinton")) %>%
  unnest_tokens(word, Text) %>%
  anti_join(stop_words) %>%
  inner_join(get_sentiments("bing")) %>%
  group_by(sentiment, Speaker) %>%
  count(word) %>%
  top_n(10) %>%
  ggplot(., aes(reorder(word, n), n, fill = Speaker)) +
  geom_col(show.legend = T) +
  coord_flip() +
  facet_wrap(~sentiment, scales = "free_y") +
  xlab("Words") +
  ylab("frequency") +
  ggtitle("Word Usage") +
  theme(plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
        axis.title.x = element_text(face = "bold", size = 13),
        axis.title.y = element_text(face = "bold", size = 13))
```



Finally, lets create a **wordcloud** of these words, visualizing positive vs negative

```
debate %>%
  filter(Speaker %in% c("Trump","Clinton")) %>%
  unnest_tokens(word, Text) %>%
  mutate(word = gsub("problems", "problem", word)) %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment) %>%
  acast(word~sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(color = c("#1b2a49", "#00909e"),
                   max.words = 100)
```


```
# Note the acast function is from the reshape2 package
# Functions such as comparison.cloud() require you to turn the data frame into a matrix with reshape2's
```