Fernando Cavina

4 – Divisadero

A3: Business Insight Report

The following analysis aims to compare news published by press channels such as CNN, New York Times and Financial Times, with other media coming from Fox News, about the Democratic primary elections, which started last week, and its problems. As it is popularly known, the Fox conglomerate is generally more of a Republican bias when it issues political news and opinions. This comparison aims to identify keywords and feelings that present a clear difference in approach between the two types of media present in the United States.

Initially, I created folders, one containing three publications coming from channels that generally have left wing opinions (CNN, New York Times and Financial Times), and the other containing two Fox News publications, all about the same topic. Then, both folders were uploaded to R Studio and two data frames were created, so that the content of all news could be analyzed in two separate ways. Then, I tokenized the data, removed the stop words, numbers and expressions that did not help in understanding the content of the articles.

Additionally, I built plots containing the token frequency for each group of posts. This activity resulted in a list of the most repeated words in each group, making it possible to identify some important differences between the articles pro Democrats, and the others published by Fox News. For example, while the first plot contains the repetition of words like "candidates", "democrats" and "leading", words in which it makes sense to be very present, the second (referring to Fox News) shows, in addition to these expected words, other like "disaster" and even the name of President "Trump" many times. Therefore, this first analysis helps us to conclude that media broadcasted by Fox News tend to be more negatively biased when dealing with the Democratic party.

Then, build the correlogram, in order to then build the plot containing this information. From this, we see that the correlogram reinforces the idea that Fox most often uses adjectives of a derogatory character when it comes to the Democratic party.

Finally, I built a Pizza Chart bringing the Sentiment Analysis present in both groups of articles. The results were very positive and bring a confirmation of what was expected through previous analyzes. For example, when it’s about Anger, the pro Democrats group uses softer words, such as "confusion", "upset", "Revolution", "chaos", "anxiety" and "struggle". However, the group containing Fox News, is more adept at more derogatory and heavy words, such as "incompetent", "humiliate", "frustrated" and "threat". Also, when it’s about Trust, the first group treats what happened as an "accident", however, the second group sees it as a "disaster". Finally, when we analyze the item Disgust, we see that the first group uses words like "uncertain", "discontent" and "delay", while the second group opts for tougher expressions, such as "punished", "terrible", "hell" and even "illegal".

It is also important to emphasize that the Sentiment Analysis model is not perfect and also has errors. We can notice that in the chart referring to the media for Democrats, there is the use of the words "surprise" characterized as Joy, when in fact this expression is used in the text with a negative connotation. Therefore, we can conclude that this model, although very accurate, can present errors.

From the analyzes carried out, I concluded that the polarization is extremely present in the media in the United States and it can strongly affect the 2020 Presidential elections. In this example, while one part of the press chooses to mitigate mistakes made by a certain party, the other chooses the opposite side and uses harsh and extreme words in order to criticize a certain attitude of a political party.

If one of those broadcast companies invests more in targeting content to customer that are willing to believe in a sensationalist narrative, it can influence in who will be the next present. Therefore, the government should be aware of this problem and try to avoid this kind of influence through the news and social media, just like it happened four years ago. It's necessary to understand the media and its role in a country as polarized as the United States, and the use of Correlogram and Sentiment analyzes in R Studio are fundamental to distinguish which side each player is in.

Sources

Article 1
Jasmine C. Lee, Annie Daniel, Rebecca Lieberman, Blacki Migliozzi, Alexander Burns and Sarah Almukhtar (Feb. 7, 2020). *Which Democrats Are Leading the 2020 Presidential Race?*
https://www.nytimes.com/interactive/2020/02/07/us/elections/democratic-polls.html

Article 2
Ronald Brownstein (Feb. 5, 2020) *The Iowa muddle isn't over, even now that the first results are in*
https://www.cnn.com/2020/02/05/politics/iowa-caucuses-candidates-voters-support-divided/index.html

Article 3
Anne Shreiner (Feb. 5 2020) *The Democrats' Iowa app fiasco was an accident waiting to happen*
https://www.ft.com/content/7ebde9f8-477b-11ea-aee2-9ddbdc86190d

Article 4
Tucker Carlson (Feb. 5 2020) *Tucker Carlson: Democrats screw up the Iowa caucuses, then blame the voters. They truly are a disaster*
https://www.foxnews.com/opinion/tucker-carlson-democrats-iowa-caucuses-disaster-voters

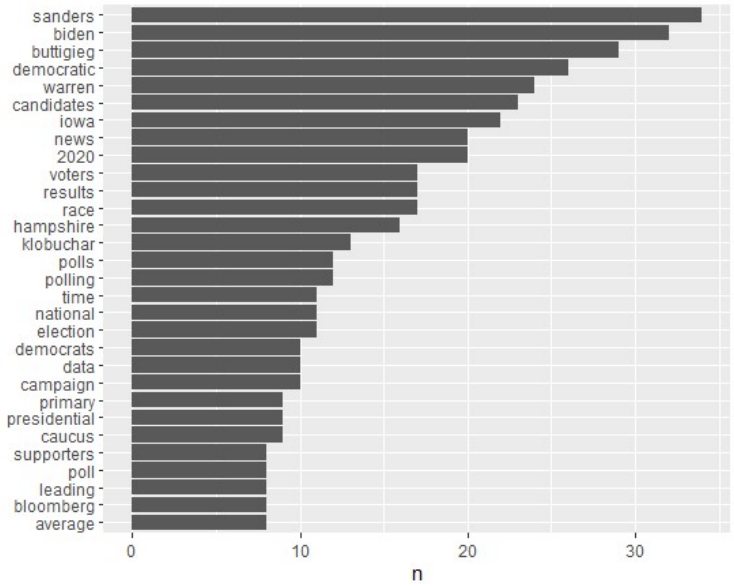
Article 5
Gregg Re, Allie Raffa (Feb. 4 2020) *Iowa Caucus vote totals delayed amid 'inconsistencies'; campaigns lash out at 'crazy' state party*
https://www.foxnews.com/politics/iowa-caucuses-currently-a-wide-open-race-with-voting-underway-in-pivotal-contest

R code

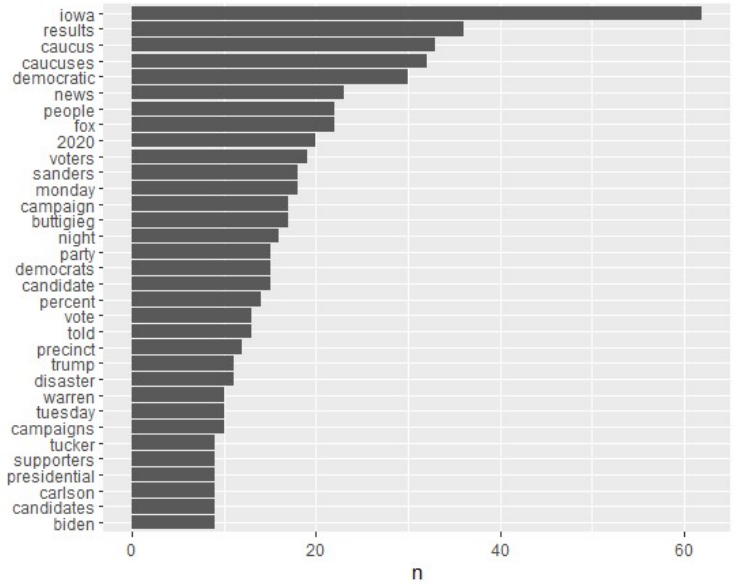
```
#installing the necessary libraries
library(pdftools)
library(tm)
library(dplyr)
library(tidytext)
library(tidyverse)
library(ggplot2)
library(scales)
library(textreadr)

setwd("C:/Users/ferna/Documents/Hult docs/MBAN/Text Analytics")
Other_Medias <- read_document(file="Other Medias/CNN.txt")
OM1_df <- as.data.frame(Other_Medias)
Other_Medias <- read_document(file="Other Medias/New York Times.txt")
OM2_df <- as.data.frame(Other_Medias)
Other_Medias <- read_document(file="Other Medias/Financial Times.txt")
OM3_df <- as.data.frame(Other_Medias)

Other_Medias <- rbind(OM1_df,OM2_df,OM3_df)
Other_Medias$Other_Medias <- as.character(Other_Medias$Other_Medias)

Fox_News <- read_document(file="Fox News/Fox News.txt")
FN1_df <- as.data.frame(Fox_News)
Fox_News <- read_document(file="Fox News/Fox News2.txt")
FN2_df <- as.data.frame(Fox_News)

Fox_News <- rbind(FN1_df,FN2_df)
Fox_News$Fox_News <- as.character(Fox_News$Fox_News)

#creating a corpus
OMcorp <- VCorpus(VectorSource(Other_Medias))
FNcorp <- VCorpus(VectorSource(Fox_News))

#creating a Term-Document-Matrices
OM.tdm <- TermDocumentMatrix(OMcorp,
                      control =
                       list(removePunctuation = TRUE,           #removing punctuation
                          stopwords = TRUE,                  #removing stopwords
                          tolower = TRUE,                   #converting to lowercase
                          removeNumbers = TRUE,              #removing numbers
                          bounds = list(global = c(3, Inf))))    #only words that appear more than 3 times

FN.tdm <- TermDocumentMatrix(FNcorp,
```

```
                  control =
                   list(removePunctuation = TRUE,           #removing punctuation
                      stopwords = TRUE,                  #removing stopwords
                      tolower = TRUE,                   #converting to lowercase
                      removeNumbers = TRUE,              #removing numbers
                      bounds = list(global = c(3, Inf))))    #only words that appear more than 3 times
inspect(OM.tdm)
inspect(FN.tdm)


#tokenize the data & remove stop words
OM_token <- Other_Medias %>%
  unnest_tokens(word, Other_Medias) %>%
  anti_join(stop_words) %>%
  count(word, sort=TRUE)
OM_token

                  # A tibble: 1,036 x 2
                     word           n
                     <chr>      <int>
                   1 sanders       34
                   2 biden         32
                   3 buttigieg     29
                   4 democratic    26
                   5 warren        24
                   6 candidates    23
                   7 iowa          22
                   8 2020          20
                   9 news          20
                  10 â              19
                  # ... with 1,026 more rows


FN_token <- Fox_News %>%
  unnest_tokens(word, Fox_News) %>%
  anti_join(stop_words) %>%
  count(word, sort=TRUE)
FN_token

                  # A tibble: 1,105 x 2
                     word           n
                     <chr>      <int>
                   1 iowa           62
                   2 results        36
                   3 caucus         33
                   4 caucuses       32
                   5 democratic     30
                   6 news           23
                   7 fox            22
                   8 people         22
                   9 2020           20
                  10 â              19
                  # ... with 1,095 more rows
```

```
# tokenizing
cust_stop <- data_frame(word=c("7","6","8","20","2","1","15","4","5","3","23","app","ng","â","de","0","10","jan","feb"),
                        lexicon=rep("cust", each=19))

OM_token <- Other_Medias  %>%
  unnest_tokens(word,Other_Medias) %>%
  anti_join(stop_words) %>%
  anti_join(cust_stop) %>%
  count(word, sort = T)

cust_stop <- data_frame(word=c("7","6","8","20","2","1","15","4","5","3","23","es","â","feb","idp","er","al","pr","del","ent","iow","ar","ults","res","ay","ow"),
                        lexicon=rep("cust", each=26))

FN_token <- Fox_News  %>%
  unnest_tokens(word,Fox_News) %>%
  anti_join(stop_words) %>%
  anti_join(cust_stop) %>%
  count(word, sort = T)

#plotting the frequency of tokens in all postings
OM_freq <- OM_token %>%
  filter(n > 7) %>%                    #we need this to eliminate all the low count words
  mutate(word = reorder(word,n )) %>%
  ggplot(aes(word, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()

print(OM_freq)
```



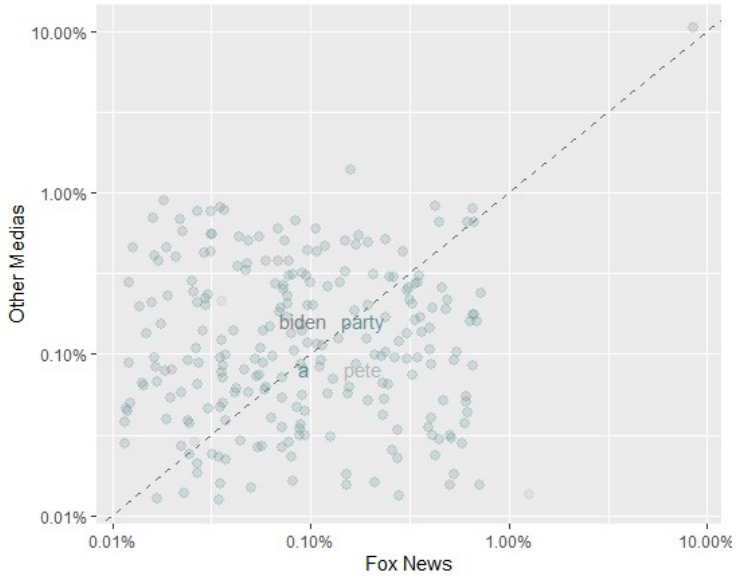

```
FN_freq <- FN_token %>%
  filter(n > 8) %>%                    #we need this to eliminate all the low count words
  mutate(word = reorder(word,n )) %>%
  ggplot(aes(word, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()

print(FN_freq)
```



```
#Preparing Correlograms by mutating to proportions
proportions <- bind_rows(mutate(OM_token, media = "Other Medias"),
                         mutate(FN_token, media = "Fox News")
)%>%                                    #closing bind_rows
  mutate   (word = str_extract(word, "[a-z']+")) %>%
  count    (media, word) %>%
  group_by (media) %>%
  mutate   (proportion = n/sum(n))%>%
  select   (-n) %>%
  spread   (media, proportion) %>%
  gather   (media, proportion, `Fox News`)
View(proportions)
```

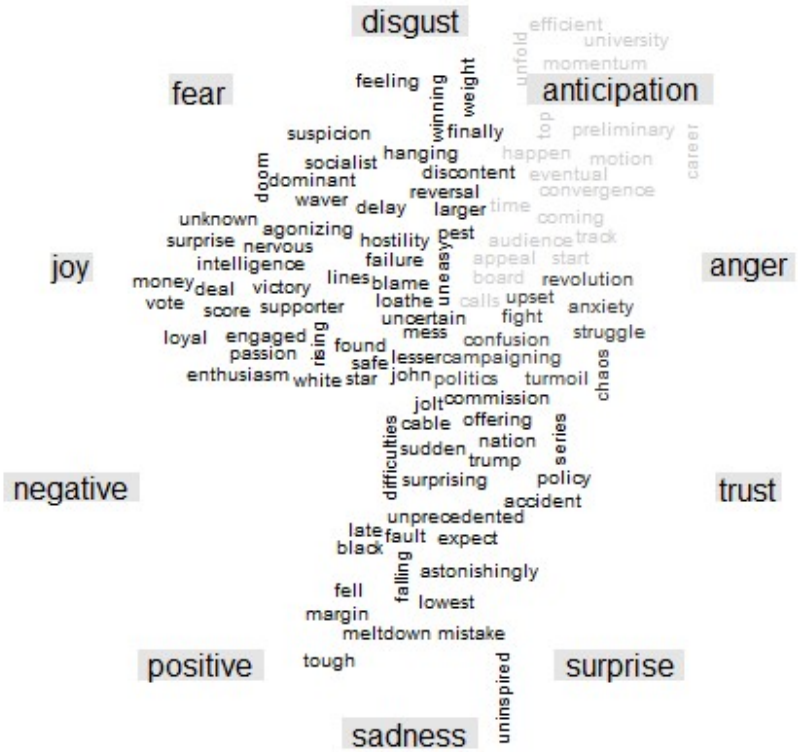
word	Other Medias	media	proportion	
1	a	0.0009813543	Fox News	0.0009174312
2	abc	0.0009813543	Fox News	*NA*
3	ability	0.0009813543	Fox News	0.0009174312
4	abroad	*NA*	Fox News	0.0009174312
5	accepted	*NA*	Fox News	0.0009174312
6	accepting	0.0009813543	Fox News	*NA*
7	accident	0.0009813543	Fox News	*NA*
8	account	*NA*	Fox News	0.0009174312
9	accumulating	*NA*	Fox News	0.0009174312

```
#Plotting the Correlogram
ggplot(proportions, aes(x     = proportion, y = `Other Medias`,
                        color = abs(`Other Medias`- proportion)))+
  geom_abline(color = "grey40", lty = 2)+
  geom_jitter(alpha = .2, size = 2.5, width   = 0.9, height = 0.9)+
  geom_text(aes(label  = word), check_overlap = TRUE, vjust = 1.3) +
  scale_x_log10(labels = percent_format())+
  scale_y_log10(labels = percent_format())+
  scale_color_gradient(limits = c(0,0.001), low = "darkslategray4", high = "gray75")+
  theme(legend.position = "none")+
  labs(y = "Other Medias", x = "Fox News")
```



#Sentiment Analysis with the NRC library - creating a "Pizza Chart"

```
OM_token %>%
 inner_join     (get_sentiments("nrc")) %>%
 count          (word, sentiment, sort=TRUE) %>%
 acast          (word ~sentiment, value.var="n", fill=0) %>%
 comparison.cloud(colors = c("grey20", "gray80"),
                  max.words=100,
                  scale=c(0.6,0.6),
                  fixed.asp = TRUE,
                  title.size = 1)
```



```
FN_token %>%
 inner_join     (get_sentiments("nrc")) %>%
 count          (word, sentiment, sort=TRUE) %>%
 acast          (word ~sentiment, value.var="n", fill=0) %>%
 comparison.cloud(colors = c("grey20", "gray80"),
                  max.words=100,
                  scale=c(0.6,0.6),
                  fixed.asp = TRUE,
                  title.size = 1)
```