Mouhamet Ndiaye

Class: Text Analytics

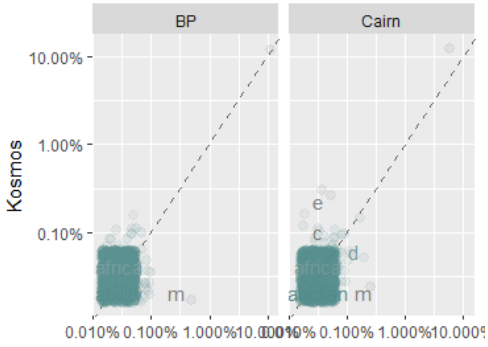
Prof. Thomas Kurnicki

02/12/2020

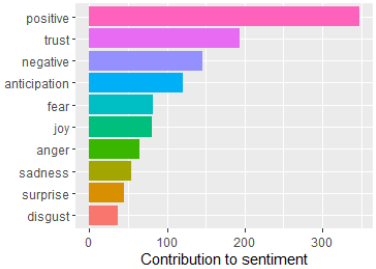
Business Insight Report

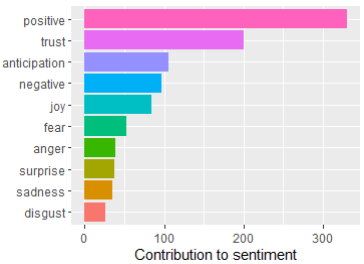
Introduction: Cairn Energy PLC, Kosmos Energy Ltd, and British PLC are independent/small, medium size, and multinational oil companies respectively. In 2016, they discovered large amounts of offshore hydrocarbon reservoirs in Senegal, my home country. In 2018, I conducted a media content analysis of the public debate around these discoveries by analyzing the government and civil society communication patterns. I saw this project as an opportunity to analyze the third stakeholder; the oil and gas companies mentioned above. In hopes of analyzing their communication frameworks, I chose to run a sentiment analysis on their 2018 Corporate Social Responsibility Reports. It is against this backdrop that I provide a word correlogram, sentiment analysis, and word cloud to derive business insights.

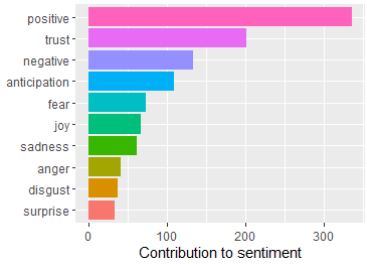
Correlogram: In this correlogram, we can see that the words are evenly and similarly distributed across the three different companies. In fact, there is a 99% correlation between the two reports relative to Kosmos and there's a 95% confidence interval indicating the accuracy of the hypothesis that there is a correlation. The graph below clearly demonstrates synchrony among these companies.



Sentiment Analysis: Below are graphs of the sentiment analysis for all the three reports starting with BP to the far left, Kosmos in the middle, and Cairn to the right. They all display a high level of trust and anticipation once more showing that the aim of their CSR reports is to establish trust and gain support from local, regional, and international communities. It is important to mention that there is both joy and fear present in the wake of oil and gas discoveries alluding to the natural resource curse.







Word Cloud: In this section, we can see the word clouds for Cairn to the left, and Kosmos to the right. The insight that we get here is that corruption, bribery, abuses and many other human right violations are rampant in the oil and gas industry. These negative words on both Kosmos & Cairn's 2018 CSR report demonstrates that they acknowledge the reality of the industry, while the positive words such as advocates, beneficial, capability, clean and so on highlights their efforts and concerns for a safe, reliable, and transparent management of their operations.

Just like the Cairn & Kosmos word clouds show, the one below demonstrates British Petroleum's CSR report. BP has words such as degradation, complaints, break, conflicts, death etc. These words, among many in their report shows the similar concerns that most oil and gas companies have. In other words, there is no doubt about the lack of transparency in the extractive industry and all these three different companies demonstrated it through their reports. The simple fact that these words are there shows that they need to be addressed, which means that there is a problem. The insight here can very well be that BP is speaking out against these acts. Nonetheless, the appearance of these words across the dataset indicates that oil companies try to communicate online to gain support at he local, regional, and international level while portraying themselves as trustworthy and reliable.

Conclusion: Upon completing this report, I came to the realization that the word patterns can be interpreted differently. However, it is important to not downplay the similarities of communication themes that emerge from these three companies of different sizes. Ultimately, oil and gas companies allocate a lot of resources to communicate with host communities, civil society groups, governments, and international stakeholders. This is done through major communication campaigns and annual reports as well as CSR reports, which I have shown all along. In sum, the insights here is that the extractive industry, now more than ever needs to be transparent. Furthermore, the correlograms, sentiment graphs, and word clouds above demonstrate a sense of communication strategy whereby oil and gas companies, irrespective of size utilize their annual CSR reports to display a positive light on their reputation, and operational activities.

The overwhelmingly positive sentiments shown by the sentiment analysis graph serves as a proof. The positive words, shown across the graphs also serves as proof that there is a clear intent of passing a positive message to gain support and establish trust. However, the negative words not only demonstrate the hostile nature of the industry, but they also show that oil companies acknowledge it. Whether or not these reports are accurate is another story. It might just be a procedural activity that they are required to publish. The true question here is how trustworthy, ethical, and transparent are they? I hope this analysis provided insights on the communication frames employed by independent as well as multinational oil companies to different communities. Choosing a small, medium, and large company was intentional as it provided a wider representation of the global oil and gas industry.

Bibliography

Energy, C. (2018). *Working Responsibly to Create Value: Cairn Energy PLC 2018 COrporate Responsibility Report.*

Energy, K. (2018). *Corporate Responsibility Report.*

Petroleum, B. (2018). *Responding to the dual challenge: BP Sustainability Report 2018.*

Code Input

```
install.packages("shapeR")
install.packages("pdftools")
install.packages("tm")
install.packages("dplyr")
install.packages("stringr")
install.packages("tidytext")
install.packages("ggplot2")
install.packages("reshape2")
install.packages("wordcloud")
install.packages("tidyverse")
install.packages("textreadr")
install.packages("textshape")
install.packages("scales")
install.packages("tidyr")
install.packages("textdata")
install.packages("plotly")
install.packages("RColorBrewer")
library(shapeR)
library(pdftools)
library(tm)
library(NLP)
library(dplyr)
library(stringr)
library(tidytext)
library(ggplot2)
library(reshape2)
library(RColorBrewer)
library(wordcloud)
```

```
library(shapeR)
library(tidyverse)
library(textreadr)
library(textshape)
library(scales)
library(tidyr)
library(textdata)
library(plotly)
################################################################################################################################
setwd("C:/Users/nabic/Desktop/articles") #PDF files location
nm <- list.files(path="C:/Users/nabic/Desktop/articles")# PDF file storage
################################################################################################################################
#######CREATING A DATA frame for the reports
################################################################################
################################################################################################################################
BP_1<-as.data.frame(pdf_text('British Petroleum Report.pdf'))%>%mutate(name='BP')
names(BP_1)[1]<-"text"
view(BP_1)
################################################################################################################################


library(pdftools)#call this library to read text
library(tm)
setwd("C:/Users/nabic/Desktop/articles")
nm <- list.files(path="/Users/nabic/Desktop/articles")
```

```
##############################################################################

#######CREATING A DATA FRAME for
KOSMOS#####################################################

##############################################################################

KR_2<-as.data.frame(pdf_text('Kosmos Report.pdf'))%>%mutate(name='Kosmos')

names(KR_2)[1]<-"text"

view(KR_2)


##############################################################################
####

setwd("C:/Users/nabic/Desktop/articles") #where are your PDF files

nm <- list.files(path="/Users/nabic/Desktop/articles")#were are your PDF files stored?


##############################################################################
##

#######CREATING A DATA FRAME FOR
Cairn#####################################################

##############################################################################
###3

CRN_3<-as.data.frame(pdf_text('Cairn Report.pdf'))%>%mutate(name='Cairn')

names(CRN_3)[1]<-"text"

view(CRN_3)

##############################################################################
######################

### creating my stop word ###

Stuup_word <- data_frame(word=c("0",3","4","8","9","10""2018""oil""gas"),

lexicon=rep("custom",each=9)

#closing data_frame

print(Stuup_word)
```

```
#####################################################
#Removal of stopwords and count of word frequency
#####################################################
data(stop_words)
frequenciesStuup_word<- Stuup_word  %>%
unnest_tokens(word, text) %>%
anti_join(stop_words) %>% #Here tokens are removed
anti_join(Stuup_word) %>% #Here custom words are removed
count(word, sort=TRUE)
##################TOKENIZING THE CSR REPORTS#####################################################
#################################################################################################
#####################
BP_1$text<-as.character(BP_1$text)
tidy_BP_1<-BP_1%>%
unnest_tokens(word, text) %>%
anti_join(stop_words,by = "word") %>%
count(word, sort=T)

View(tidy_BP_1)

KR_2$text<-as.character(KR_2$text)
tidy_KR_2<-KR_2%>%
unnest_tokens(word, text) %>%
anti_join(stop_words,by = "word")%>%
count(word, sort=T)

View(tidy_KR_2)
```

```
CRN_3$text<-as.character(CRN_3$text)

tidy_CRN_3<-CRN_3%>%

unnest_tokens(word, text) %>%

anti_join(stop_words,by = "word")%>%

count(word, sort=T)


View(tidy_CRN_3)


################################################################################
######################

##### TOKEN FREQUENCY HISTOGRAMS FOR MY DATA FRAMES
############################################################

################################################################################
########################


#BRITISH PETROLEUM 2018 CORPORATE SOCIAL RESPONSIBILITY REPORT

library(ggplot2)

freq_hist <-tidy_BP_1%>%

#anti_join(stop_words) %>%

#count(word, sort=TRUE) %>%

filter(n>100)%>%

ggplot(aes(word,n,fill= sentiment))+

geom_col()+

xlab=(NULL)+

coord_flip()


freq_hist


#KOSMOS ENERGY 2018 CORPORATE SOCIAL RESPONSIBILITY REPORT

library(ggplot2)
```

```
freq_hist <-tidy_KR_2%>%
#anti_join(stop_words) %>%
#count(word, sort=TRUE) %>%
filter(n>75)%>%
ggplot(aes(word, n))+
geom_col()+
xlab(NULL)+
coord_flip()

freq_hist

#CAIRN ENERGY 2018 CORPORATE SOCIAL RESPONSIBILITY REPORT
library(ggplot2)
freq_hist <-tidy_CRN_3%>%
#anti_join(stop_words) %>%
#count(word, sort=TRUE) %>%
filter(n>80)%>%
ggplot(aes(word, n))+
geom_col()+
xlab(NULL)+
coord_flip()

freq_hist

################################################################################
########################

##################RUNNING A CORRELOGRAM WITH MY THREE DATAFRAME FOR CORRELATION###################################

library(tidyr)
```

```
frequency<-bind_rows(mutate(tidy_BP_1,file="BP"),
                     mutate(tidy_KR_2,file="Kosmos"),
                     mutate(tidy_CRN_3,file="Cairn"))%>%


mutate(word=str_extract(word, "[a-z']+")) %>%
count(file, word) %>%
group_by(file) %>%
mutate(proportion = n/sum(n))%>%
select(-n) %>%
spread(file, proportion) %>%
gather(file, proportion, `BP`, `Cairn`)

#plotting the correlograms:

library(scales)

ggplot(frequency, aes(x=proportion, y=`Kosmos`,
                      color = abs(`Kosmos`- proportion)))+
geom_abline(color="grey40", lty=2)+
geom_jitter(alpha=.1, size=2.5, width=0.3, height=0.3)+
geom_text(aes(label=word), check_overlap = TRUE, vjust=1.5) +
scale_x_log10(labels = percent_format())+
scale_y_log10(labels= percent_format())+
scale_color_gradient(limits = c(0,0.001), low = "darkslategray4", high = "gray75")+
facet_wrap(~file, ncol=2)+
theme(legend.position = "none")+
labs(y= "Kosmos", x=NULL)
```

```
#Taking a look at correlation coefficients
cor.test(data=frequency[frequency$file == "BP",],
~proportion + `Kosmos`)

cor.test(data=frequency[frequency$author == "Cairn",],
~proportion + `Kosmos`)

################################################################################
##RUNNING A SENTIMENT ANALYSIS#################################################
################################################################################
library(tidytext)
get_sentiments('afinn') # Show example of the table

# pulling in sentiment for these 3 tokenized datasets
tidy_BP_1%>%
inner_join(get_sentiments("afinn"))%>%
#if you remove the group_by it will calculate sentiment for all the data
summarise(sentiment=sum(value)) %>%
mutate(method="AFINN")



tidy_KR_2 %>%
inner_join(get_sentiments("afinn"))%>%
summarise(sentiment=sum(value)) %>%
mutate(method="AFINN")

tidy_CRN_3 %>%
inner_join(get_sentiments("afinn"))%>%
```

```
summarise(sentiment=sum(value)) %>%
mutate(method="AFINN")

##let's see most negative tokens in the BP report

tidy_BP_1_sentiment <- tidy_BP_1 %>%
 inner_join(get_sentiments("nrc")) %>%
count(sentiment, sort=T) %>%
mutate(sentiment = reorder(sentiment, n))


print(tidy_BP_1_sentiment)

tidy_BP_1_sentiment %>%
#group_by(sentiment) %>%
#top_n(10) %>%
#ungroup() %>%
#mutate(word=reorder(word, n)) %>%
ggplot(aes(sentiment, n, fill=sentiment)) +
geom_col(show.legend = FALSE) +
#facet_wrap(~sentiment, scales = "free_y")+
labs(y="Contribution to sentiment", x=NULL)+
coord_flip()
#the most positive and most negative tokens in the Kosmos report
tidy_KR_2_sentiment <- tidy_KR_2 %>%
 inner_join(get_sentiments("nrc")) %>%
count(sentiment, sort=T) %>%
mutate(sentiment = reorder(sentiment, n))
```

```
print(tidy_KR_2_sentiment)

tidy_KR_2_sentiment %>%
 #group_by(sentiment) %>%
 #top_n(10) %>%
 #ungroup() %>%
#mutate(word=reorder(word, n)) %>%
ggplot(aes(sentiment, n, fill=sentiment)) +
geom_col(show.legend = FALSE) +
#facet_wrap(~sentiment, scales = "free_y")+
labs(y="Contribution to sentiment", x=NULL)+
coord_flip()
##the most positive and most negative tokens in the Cairn report
tidy_CRN_3_sentiment <- tidy_CRN_3 %>%
inner_join(get_sentiments("nrc")) %>%
count(sentiment, sort=T) %>%
mutate(sentiment = reorder(sentiment, n))

print(tidy_CRN_3_sentiment)

tidy_CRN_3_sentiment %>%
 #group_by(sentiment) %>%
 #top_n(10) %>%
 #ungroup() %>%
#mutate(word=reorder(word, n)) %>%
ggplot(aes(sentiment, n, fill=sentiment)) +
geom_col(show.legend = FALSE) +
```

```
#facet_wrap(~sentiment, scales = "free_y")+
labs(y="Contribution to sentiment", x=NULL)+
coord_flip()
##############################################################################
###RUNNING A WORD CLOUD FRAME WORK#############################################
#install.packages("wordcloud")
library(wordcloud)



tidy_BP_1_sentiment <- tidy_BP_1 %>%
inner_join(get_sentiments("bing")) %>%
count(word, sentiment, sort=T)%>%
acast(word~sentiment,value.var = "n",fill=0)%>%
comparison.cloud(colors=c("red","darkgreen"),
                 max.words=500, scale=c (1,1),
                 fixed.asp=TRUE,title.size=1)

tidy_KR_2_sentiment <- tidy_KR_2 %>%
inner_join(get_sentiments("bing")) %>%
count(word, sentiment, sort=T)%>%
acast(word~sentiment,value.var = "n",fill=0)%>%
comparison.cloud(colors=c("red","blue"),
                 max.words=500, scale=c (1,1),
                 fixed.asp=TRUE,title.size=1)

tidy_CRN_3_sentiment <- tidy_CRN_3 %>%
inner_join(get_sentiments("bing")) %>%
count(word, sentiment, sort=T)%>%
```

```
acast(word~sentiment,value.var = "n",fill=0)%>%

comparison.cloud(colors=c("maroon","turquoise"),

max.words=500, scale=c (1,1),

fixed.asp=TRUE,title.size=1)

warnings()
```

Output

```
package 'RColorBrewer' successfully unpacked and MD5 sums checked
Error in install.packages : ERROR: failed to lock directory 'C:\Users\nabic\D
ocuments\R\win-library\3.6' for modifying
Try removing 'C:\Users\nabic\Documents\R\win-library\3.6/00LOCK'
> library(shapeR)
> library(pdftools)
> library(tm)
> library(NLP)
> library(dplyr)

> library(stringr)
> library(tidytext)
> library(ggplot2)

> library(reshape2)
> library(RColorBrewer)
> library(wordcloud)
> library(shapeR)
> library(tidyverse)
-- Attaching packages --------------------------------------- tidyverse 1.3.0
--
v tibble  2.1.3     v purrr   0.3.3
v tidyr   1.0.2     v forcats 0.4.0
v readr   1.3.1
-- Conflicts ------------------------------------------ tidyverse_conflicts()
--
x ggplot2::annotate() masks NLP::annotate()
x dplyr::filter()     masks stats::filter()
x dplyr::lag()        masks stats::lag()
> library(textreadr)
> library(textshape)

> library(scales)


> library(tidyr)
> library(textdata)
> library(plotly)


> ##############################################################################
##########################################
> setwd("C:/Users/nabic/Desktop/articles") #PDF files location
> nm <- list.files(path="C:/Users/nabic/Desktop/articles")# PDF file storage
```

```
> ##########################################################################
############################
> #######CREATING A DATA frame for the reports ##############################
########################################
> ##########################################################################
########################
> BP_1<-as.data.frame(pdf_text('British Petroleum Report.pdf'))%>%mutate(name
='BP')
> names(BP_1)[1]<-"text"
> view(BP_1)
> library(pdftools)#call this library to read text
> library(tm)
> setwd("C:/Users/nabic/Desktop/articles")
> nm <- list.files(path="/Users/nabic/Desktop/articles")
> ##########################################################################
#########
> #######CREATING A DATA FRAME for KOSMOS#####################################
################
> ##########################################################################
#########
> KR_2<-as.data.frame(pdf_text('Kosmos Report.pdf'))%>%mutate(name='Kosmos')
> names(KR_2)[1]<-"text"
> view(KR_2)
> ##########################################################################
#############
> setwd("C:/Users/nabic/Desktop/articles") #where are your PDF files
> nm <- list.files(path="/Users/nabic/Desktop/articles")#were are your PDF fi
les stored?
> ##########################################################################
############
> #######CREATING A DATA FRAME FOR Cairn#####################################
###############
> ##########################################################################
#############3
> CRN_3<-as.data.frame(pdf_text('Cairn Report.pdf'))%>%mutate(name='Cairn')
> names(CRN_3)[1]<-"text"
> view(CRN_3)
> ##########################################################################
##############################
> ### creating my stop word ###
> Stuup_word <- data_frame(word=c("0",3","4","8","9","10""2018""oil""gas"),
Error: unexpected string constant in "Stuup_word <- data_frame(word=c("0",3",
""
>   lexicon=rep("custom",each=9)
>   #closing data_frame
> print(Stuup_word)
Error in print(Stuup_word) : object 'Stuup_word' not found
>
> ##################################################
> #Removal of stopwords and count of word frequency
> ##################################################
> data(stop_words)
> frequenciesStuup_word<- Stuup_word  %>%
+   unnest_tokens(word, text) %>%
+   anti_join(stop_words) %>% #Here tokens are removed
+   anti_join(Stuup_word) %>% #Here custom words are removed
+   count(word, sort=TRUE)
Error in eval(lhs, parent, parent) : object 'Stuup_word' not found
> ################TOKENIZING THE CSR REPORTS#################################
###############
> ##########################################################################
############################
> BP_1$text<-as.character(BP_1$text)
> tidy_BP_1<-BP_1%>%
```

```
+   unnest_tokens(word, text) %>%
+   anti_join(stop_words,by = "word") %>%
+   count(word, sort=T)
>
> View(tidy_BP_1)
>
> KR_2$text<-as.character(KR_2$text)
> tidy_KR_2<-KR_2%>%
+   unnest_tokens(word, text) %>%
+   anti_join(stop_words,by = "word")%>%
+   count(word, sort=T)
>
> View(tidy_KR_2)
>
> CRN_3$text<-as.character(CRN_3$text)
> tidy_CRN_3<-CRN_3%>%
+   unnest_tokens(word, text) %>%
+   anti_join(stop_words,by = "word")%>%
+   count(word, sort=T)
>
> View(tidy_CRN_3)
>
> ##########################################################################
##############################
> ##### TOKEN FREQUENCY HISTOGRAMS FOR MY DATA FRAMES ######################
##################################
> ##########################################################################
##################################
>
> #BRITISH PETROLEUM 2018 CORPORATE SOCIAL RESPONSIBILITY REPORT
> library(ggplot2)
> freq_hist <-tidy_BP_1%>%
+  #anti_join(stop_words) %>%
+  #count(word, sort=TRUE) %>%
+   filter(n>100)%>%
+   ggplot(aes(word,n,fill= sentiment))+
+   geom_col()+
+   xlab=(NULL)+
+   coord_flip()
Error in freq_hist <- tidy_BP_1 %>% filter(n > 100) %>% ggplot(aes(word,  :
  object 'freq_hist' not found
>
> freq_hist
Error: object 'freq_hist' not found
>
> #KOSMOS ENERGY 2018 CORPORATE SOCIAL RESPONSIBILITY REPORT
> library(ggplot2)
> freq_hist <-tidy_KR_2%>%
+   #anti_join(stop_words) %>%
+   #count(word, sort=TRUE) %>%
+   filter(n>75)%>%
+   ggplot(aes(word, n))+
+   geom_col()+
+   xlab(NULL)+
+   coord_flip()
>
> freq_hist
>
> #CAIRN ENERGY 2018 CORPORATE SOCIAL RESPONSIBILITY REPORT
> library(ggplot2)
> freq_hist <-tidy_CRN_3%>%
+   #anti_join(stop_words) %>%
+   #count(word, sort=TRUE) %>%
+   filter(n>80)%>%
```

```
+   ggplot(aes(word, n))+
+   geom_col()+
+   xlab(NULL)+
+   coord_flip()
>
> freq_hist
>
> ##########################################################################
##################################
> ##################RUNNING A CORRELOGRAM WITH MY THREE DATAFRAME FOR CORRELAT
ION#################################
> ##########################################################################
##############################
> library(tidyr)
> frequency<-bind_rows(mutate(tidy_BP_1,file="BP"),
+                      mutate(tidy_KR_2,file="Kosmos"),
+                      mutate(tidy_CRN_3,file="Cairn"))%>%
+
+
+   mutate(word=str_extract(word, "[a-z']+")) %>%
+   count(file, word) %>%
+   group_by(file) %>%
+   mutate(proportion = n/sum(n))%>%
+   select(-n) %>%
+   spread(file, proportion) %>%
+   gather(file, proportion, `BP`, `Cairn`)
>
> #plotting the correlograms:
>
> library(scales)
>
> ggplot(frequency, aes(x=proportion, y=`Kosmos`,
+                       color = abs(`Kosmos`- proportion)))+
+   geom_abline(color="grey40", lty=2)+
+   geom_jitter(alpha=.1, size=2.5, width=0.3, height=0.3)+
+   geom_text(aes(label=word), check_overlap = TRUE, vjust=1.5) +
+   scale_x_log10(labels = percent_format())+
+   scale_y_log10(labels= percent_format())+
+   scale_color_gradient(limits = c(0,0.001), low = "darkslategray4", high =
"gray75")+
+   facet_wrap(~file, ncol=2)+
+   theme(legend.position = "none")+
+   labs(y= "Kosmos", x=NULL)
Warning messages:
1: Removed 8209 rows containing missing values (geom_point).
2: Removed 8211 rows containing missing values (geom_text).
>
> #Taking a look at correlation coefficients
> cor.test(data=frequency[frequency$file == "BP",],
+          ~proportion + `Kosmos`)

        Pearson's product-moment correlation

data:  proportion and Kosmos
t = 1273, df = 1384, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.9995258 0.9996159
sample estimates:
      cor
0.9995733

>
> cor.test(data=frequency[frequency$author == "Cairn",],
```

```
+          ~proportion + `Kosmos`)
>
> ##############################################################################
#########
> ##RUNNING A SENTIMENT ANALYSIS#################################################
########
> ##############################################################################
######
> library(tidytext)
> get_sentiments('afinn') # Show example of the table
# A tibble: 2,477 x 2
   word       value
   <chr>      <dbl>
 1 abandon       -2
 2 abandoned     -2
 3 abandons      -2
 4 abducted      -2
 5 abduction     -2
 6 abductions    -2
 7 abhor         -3
 8 abhorred      -3
 9 abhorrent     -3
10 abhors        -3
# ... with 2,467 more rows
>
> # pulling in sentiment for these 3 tokenized datasets
> tidy_BP_1%>%
+   inner_join(get_sentiments("afinn"))%>%
+   #if you remove the group_by it will calculate sentiment for all the data
+   summarise(sentiment=sum(value)) %>%
+   mutate(method="AFINN")
Joining, by = "word"
# A tibble: 1 x 2
  sentiment method
      <dbl> <chr>
1        93 AFINN
>
>
>
> tidy_KR_2 %>%
+   inner_join(get_sentiments("afinn"))%>%
+   summarise(sentiment=sum(value)) %>%
+   mutate(method="AFINN")
Joining, by = "word"
# A tibble: 1 x 2
  sentiment method
      <dbl> <chr>
1       182 AFINN
>
> tidy_CRN_3 %>%
+   inner_join(get_sentiments("afinn"))%>%
+   summarise(sentiment=sum(value)) %>%
+   mutate(method="AFINN")
Joining, by = "word"
# A tibble: 1 x 2
  sentiment method
      <dbl> <chr>
1       106 AFINN
>
> ##let's see most negative tokens in the BP report
> tidy_BP_1_sentiment <- tidy_BP_1 %>%
+   inner_join(get_sentiments("nrc")) %>%
+   count(sentiment, sort=T) %>%
+   mutate(sentiment = reorder(sentiment, n))
```

```
Joining, by = "word"
> print(tidy_BP_1_sentiment)
# A tibble: 10 x 2
   sentiment        n
   <fct>        <int>
 1 positive       348
 2 trust          193
 3 negative       146
 4 anticipation   121
 5 fear            82
 6 joy             80
 7 anger           65
 8 sadness         54
 9 surprise        44
10 disgust         36
> tidy_BP_1_sentiment %>%
+   #group_by(sentiment) %>%
+   #top_n(10) %>%
+   #ungroup() %>%
+   #mutate(word=reorder(word, n)) %>%
+   ggplot(aes(sentiment, n, fill=sentiment)) +
+   geom_col(show.legend = FALSE) +
+   #facet_wrap(~sentiment, scales = "free_y")+
+   labs(y="Contribution to sentiment", x=NULL)+
+   coord_flip()
> #the most positive and most negative tokens in the Kosmos report
> tidy_KR_2_sentiment <- tidy_KR_2 %>%
+   inner_join(get_sentiments("nrc")) %>%
+   count(sentiment, sort=T) %>%
+   mutate(sentiment = reorder(sentiment, n))
Joining, by = "word"
> print(tidy_KR_2_sentiment)
# A tibble: 10 x 2
   sentiment        n
   <fct>        <int>
 1 positive       331
 2 trust          201
 3 anticipation   106
 4 negative        97
 5 joy             84
 6 fear            52
 7 anger           38
 8 surprise        37
 9 sadness         35
10 disgust         26
> tidy_KR_2_sentiment %>%
+   #group_by(sentiment) %>%
+   #top_n(10) %>%
+   #ungroup() %>%
+   #mutate(word=reorder(word, n)) %>%
+   ggplot(aes(sentiment, n, fill=sentiment)) +
+   geom_col(show.legend = FALSE) +
+   #facet_wrap(~sentiment, scales = "free_y")+
+   labs(y="Contribution to sentiment", x=NULL)+
+   coord_flip()
> ##the most positive and most negative tokens in the Cairn report
> tidy_CRN_3_sentiment <- tidy_CRN_3 %>%
+   inner_join(get_sentiments("nrc")) %>%
+   count(sentiment, sort=T) %>%
+   mutate(sentiment = reorder(sentiment, n))
Joining, by = "word"
> print(tidy_CRN_3_sentiment)
# A tibble: 10 x 2
   sentiment        n
```

```
   <fct>         <int>
 1 positive        336
 2 trust           201
 3 negative        133
 4 anticipation    108
 5 fear             73
 6 joy              66
 7 sadness          61
 8 anger            40
 9 disgust          37
10 surprise         33
> tidy_CRN_3_sentiment %>%
+   #group_by(sentiment) %>%
+   #top_n(10) %>%
+   #ungroup() %>%
+   #mutate(word=reorder(word, n)) %>%
+   ggplot(aes(sentiment, n, fill=sentiment)) +
+   geom_col(show.legend = FALSE) +
+   #facet_wrap(~sentiment, scales = "free_y")+
+   labs(y="Contribution to sentiment", x=NULL)+
+   coord_flip()
> ##########################################################################
######
> ###RUNNING A WORD CLOUD FRAME WORK#########################################
#######
> ##########################################################################
#########
> #install.packages("wordcloud")
> library(wordcloud)
> tidy_BP_1_sentiment <- tidy_BP_1 %>%
+   inner_join(get_sentiments("bing")) %>%
+   count(word, sentiment, sort=T)%>%
+   acast(word~sentiment,value.var = "n",fill=0)%>%
+   comparison.cloud(colors=c("red","darkgreen"),
+                    max.words=500, scale=c (1,1),
+                    fixed.asp=TRUE,title.size=1)
Joining, by = "word"
There were 50 or more warnings (use warnings() to see the first 50)
> tidy_KR_2_sentiment <- tidy_KR_2 %>%
+   inner_join(get_sentiments("bing")) %>%
+   count(word, sentiment, sort=T)%>%
+   acast(word~sentiment,value.var = "n",fill=0)%>%
+   comparison.cloud(colors=c("red","blue"),
+                    max.words=500, scale=c (1,1),
+                    fixed.asp=TRUE,title.size=1)
Joining, by = "word"
There were 50 or more warnings (use warnings() to see the first 50)
> tidy_CRN_3_sentiment <- tidy_CRN_3 %>%
+   inner_join(get_sentiments("bing")) %>%
+   count(word, sentiment, sort=T)%>%
+   acast(word~sentiment,value.var = "n",fill=0)%>%
+   comparison.cloud(colors=c("maroon","turquoise"),
+                    max.words=500, scale=c (1,1),
+                    fixed.asp=TRUE,title.size=1)
```