assignment_2_sentiment_analysis1

Marie Rivers

4/14/2022

Sentiment Analysis I

This assignment uses sentiment analysis to visualize the sentiment change over time in headlines from news articles containing the term 'IPCC'. This assignment also analyzes the change in sentiment category over time for words in news articles containing the term 'PFAS'.

```
library(tidyr) #text analysis in R
library(lubridate) #working with date data
library(pdftools) #read in pdfs
library(tidyverse)
library(tidytext)
library(here)
library(LexisNexisTools) #Nexis Uni data wrangling
library(sentimentr)
library(readr)
```

Using the "IPCC" Nexis Uni data set from the class presentation and the pseudo code we discussed, recreate Figure 1A from Froelich et al. (Date x # of 1) positive, 2) negative, 3) neutral headlines):

```
data_folder <- "/Users/marierivers/Documents/UCSB_Environmental_Data_Science/EDS_231_Text_and_Sentiment_
my_files <- list.files(pattern = ".docx", path = data_folder,
                       full.names = TRUE, recursive = TRUE, ignore.case = TRUE)

dat <- lnt_read(my_files) #Object of class 'LNT output'
# lnt_read = read in a LexisNexis file
```

```
meta_df <- dat@meta
articles_df <- dat@articles
paragraphs_df <- dat@paragraphs
```

```
dat2<- data_frame(element_id = seq(1:length(meta_df$Headline)), Date = meta_df$Date, Headline = meta_df$
paragraphs_dat <- data_frame(element_id = paragraphs_df$Art_ID, Text  = paragraphs_df$Paragraph)
dat3 <- inner_join(dat2, paragraphs_dat, by = "element_id")
```

```
mytext <- get_sentences(dat2$Headline)
sent <- sentiment(mytext)

sent_df <- inner_join(dat2, sent, by = "element_id")
```

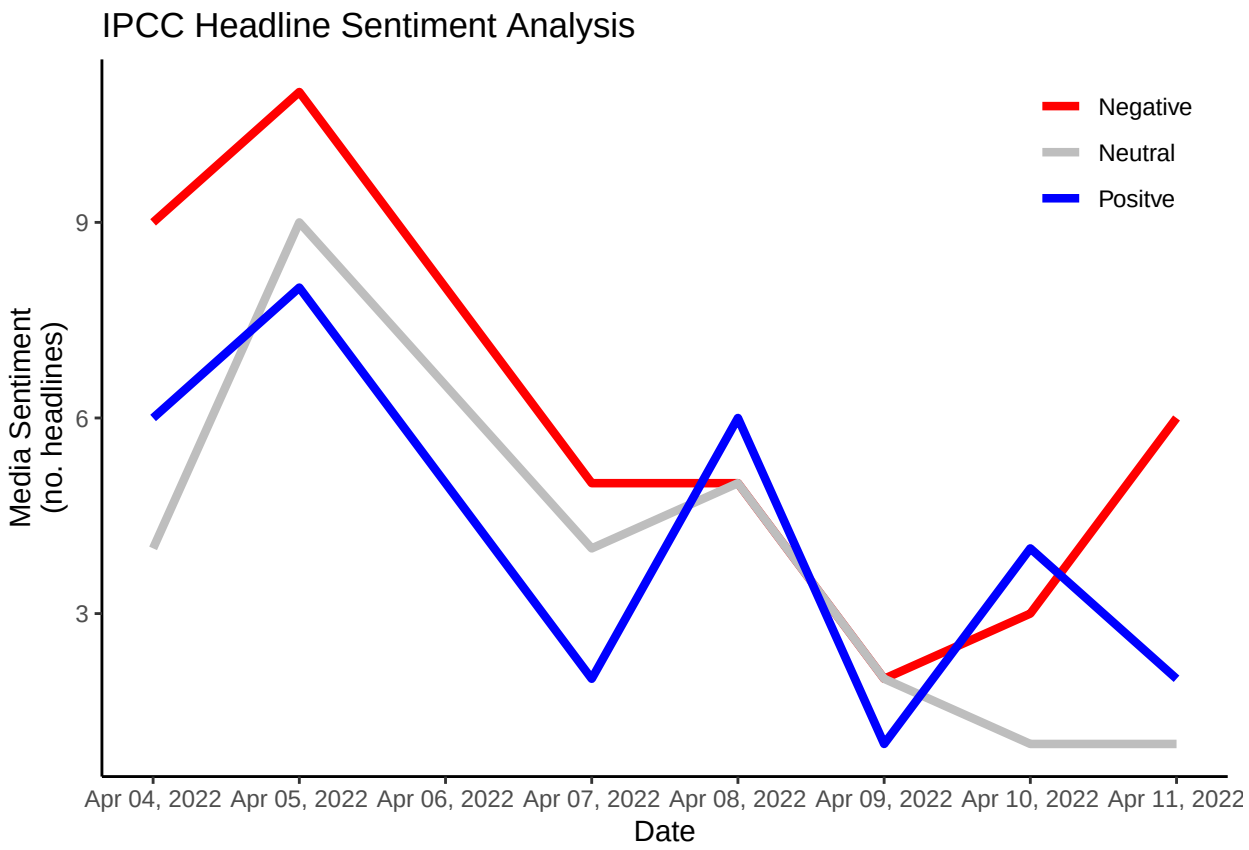
```
sentiment <- sentiment_by(sent_df$Headline)

sent_df <- sent_df%>%
  arrange(sentiment)
```

```
sent_df_summary <- sent_df %>%
  mutate(sent_category = case_when(
    sentiment < 0 ~ "negative",
    sentiment > 0 ~ "positive",
    sentiment == 0 ~ "neutral")) %>%
  group_by(Date, sent_category) %>%
  summarise(num_headlines = n())
```

```
## `summarise()` has grouped output by 'Date'. You can override using the
## `.groups` argument.
```

```
ggplot(data = sent_df_summary, aes(x = Date, y = num_headlines)) +
  geom_line(aes(color = sent_category), size = 1.5) +
  scale_color_manual(name = "",
                     values = c("red", "gray", "blue"),
                     labels = c ("Negative", "Neutral", "Positve")) +
  labs(title = "IPCC Headline Sentiment Analysis",
       y = "Media Sentiment\n(no. headlines)") +
  theme_classic() +
  theme(legend.position = c(0.9, 0.9),
        legend.background = element_blank()) +
  scale_x_date(date_labels = "%b %d, %Y",
               limits = c(as.Date("2022-04-04"), as.Date("2022-04-11")),
               breaks = "1 day")
```



Nexis Uni search of the term, 'pfas'

I started by downloading articles for the past year: from April 1, 2021 to April 1, 2022.

```
data_folder_pfas <- "/Users/marierivers/Documents/UCSB_Environmental_Data_Science/EDS_231_Text_and_Senti
my_files_pfas <- list.files(pattern = ".docx", path = data_folder_pfas,
                        full.names = TRUE, recursive = TRUE, ignore.case = TRUE)

dat_pfas <- lnt_read(my_files_pfas) #Object of class 'LNT output'
# lnt_read = read in a LexisNexis file
```

```
meta_pfas_df <- dat_pfas@meta %>%
  distinct(Headline, .keep_all = TRUE) %>% # remove duplicate headlines
  filter(!Headline %in% c("Natick water filter nearly complete; Activated carbon systemat Springvale tr
# remove headline not recognized as duplicates due to typos or differences in ' vs '

articles_pfas_df <- dat_pfas@articles
paragraphs_pfas_df <- dat_pfas@paragraphs

num_articles <- nrow(meta_pfas_df)
```

After removing duplicate headlines, this analysis uses 449 articles.

```
# headlines
dat2_pfas<- data_frame(element_id = seq(1:length(meta_pfas_df$Headline)), Date = meta_pfas_df$Date, Hea

paragraphs_dat_pfas <- data_frame(element_id = paragraphs_pfas_df$Art_ID, Text  = paragraphs_pfas_df$Pa

dat3_pfas <- inner_join(dat2_pfas,paragraphs_dat_pfas, by = "element_id") %>%
  filter(!Text %in% c("« Previous", "0 Comments", "Advertisement", "article", "Dear Abby", "Editorial",
  filter(!is.na(Date))

dat3_pfas <- dat3_pfas[!grepl("<img width=", dat3_pfas$Text),]
dat3_pfas <- dat3_pfas[!grepl("FLY FISH WINNER", dat3_pfas$Text),]

# focused on removing text that would not get removed when joining the nrc sentiments
```

NRC lexicon

```
nrc_sent <- get_sentiments('nrc') %>%
  filter(!sentiment %in% c("positive", "negative"))
```

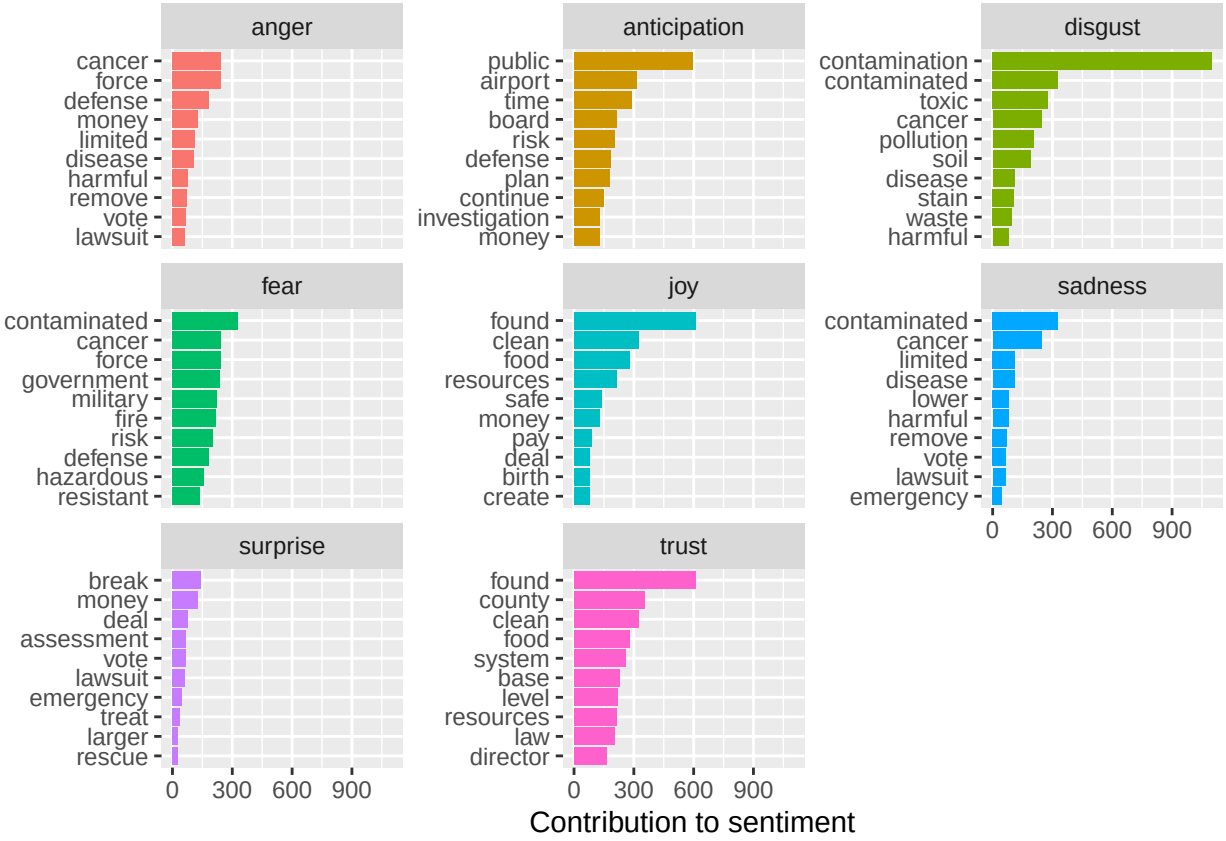
```
custom_stop_words <- bind_rows(tibble(word = c("3m"),
                                      lexicon = c("custom")),
                               stop_words)
```

```
pfas_text_words <- dat3_pfas %>%
  unnest_tokens(output = word, input = Text, token = 'words') %>%
  anti_join(custom_stop_words, by = 'word')
```

```
pfas_nrc_sent <- pfas_text_words %>%
  inner_join(nrc_sent) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

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

```
pfas_sent_plot1 <- pfas_nrc_sent %>%
  group_by(sentiment) %>%
  slice_max(n, n = 10) %>%
  ggplot(aes(x = n, y = reorder(word, n), fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(x = "Contribution to sentiment",
       y = NULL)
pfas_sent_plot1
```



This figure shows the contribution of each sentiment category based on words contained in news articles referencing pfas. The word 'contaminated' contributes most to the categories disgust, fear, and sadness. In the context of pfas, the word 'found' is usually used for finding harmful chemicals, however the NRC sentiment lexicon associates this word with the sentiments joy and trust, two emotions not felt when you find pfas contamination.

```
words_per_day <- pfas_text_words %>%
  inner_join(nrc_sent) %>%
  group_by(Date) %>%
  count(name = "words_per_day")
```

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

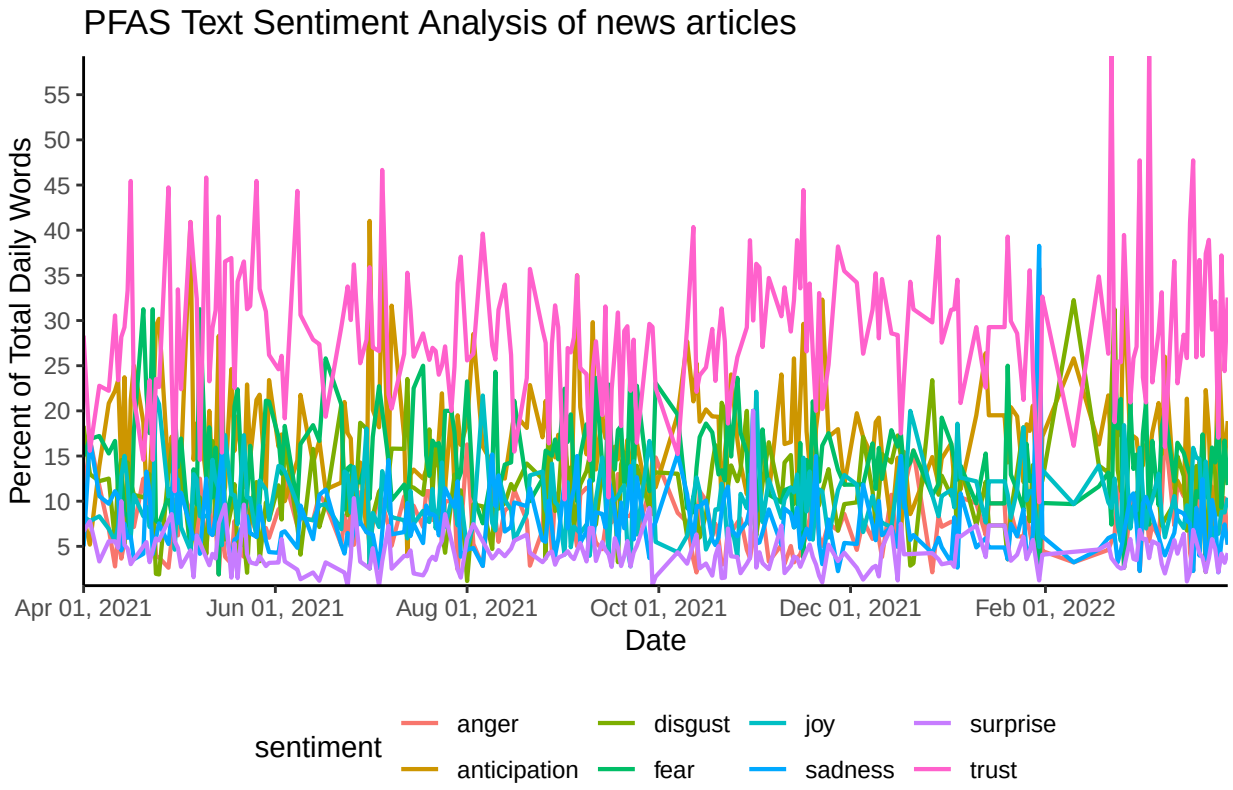
```
pfas_sent_percent <- pfas_text_words %>%
  inner_join(nrc_sent) %>%
  count(Date, sentiment) %>%
  left_join(words_per_day, by = "Date") %>%
  mutate(percent = round(((n / words_per_day) * 100), 2))
```

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

```
start_date <- min(pfas_sent_percent$Date)
end_date <- max(pfas_sent_percent$Date)

pfas_sent_plot2 <- ggplot(data = pfas_sent_percent, aes(x = Date, y = percent)) +
```

```
  geom_line(aes(color = sentiment), size = 0.75) +
  labs(title = "PFAS Text Sentiment Analysis of news articles",
       caption = "Source: Nexis Uni",
       y = "Percent of Total Daily Words") +
  theme_classic() +
  scale_x_date(date_labels = "%b %d, %Y", limits = c(start_date, end_date), expand=c(0,0), breaks = "2 m
  scale_y_continuous(breaks = seq(0, 100, by = 5), expand=c(0,0)) +
  theme(legend.position = 'bottom')
pfas_sent_plot2
```



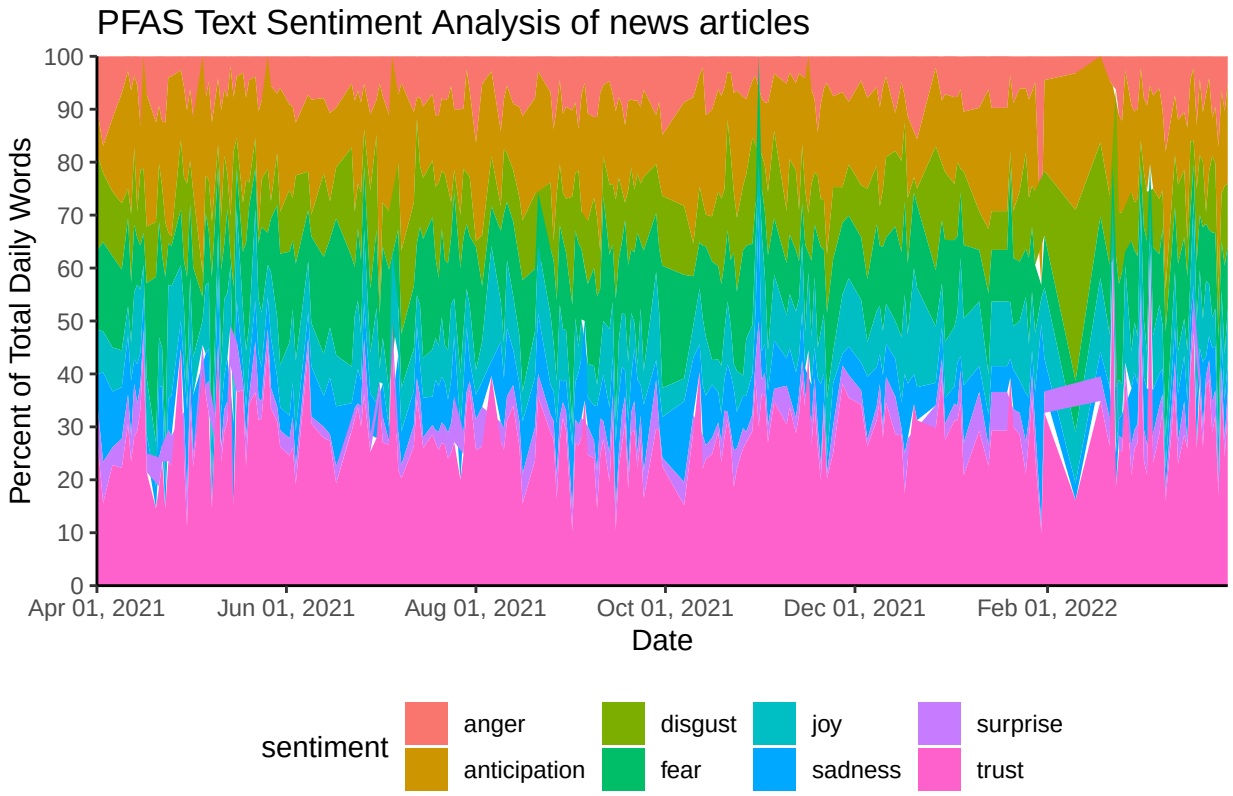
Source: Nexis Uni

Overall, trust is the most frequently occurring sentiment, however this is due to the improper association of words such as 'found', 'clean', and 'food' in the context of pfas. Articles discuses the need to clean contaminated sites and how toxic chemicals could be found in the food we grow and eat. These results illuminate the limitations of the NRC sentiment lexicon.

Overall, the distribution of emotion words does not change over time. Trust occurs most frequently followed by anticipation, fear, and sadness. Surprise is consistently the least occurring sentiment. This lack of trends may be due to the static nature of the pfas issue in the past year; pfas chemicals continue to be found in a wide range of environments, treatment capabilities are limited and/or expensive, and state/federal regulations continue to be inadequate and/or delayed by the lack of sufficient resources. Currently, the pfas problem is well documented, but little progress has been made to protect the public and environment.

```
pfas_sent_plot3 <- ggplot(data = pfas_sent_percent, aes(x = Date, y = percent, fill = sentiment)) +
  geom_area() +
  labs(title = "PFAS Text Sentiment Analysis of news articles",
       caption = "Source: Nexis Uni",
```

```
       y = "Percent of Total Daily Words") +
  theme_classic() +
  scale_x_date(date_labels = "%b %d, %Y",
               limits = c(start_date, end_date),
               expand=c(0,0),
               breaks = "2 month") +
  scale_y_continuous(breaks = seq(0, 100, by = 10), expand=c(0,0)) +
  theme(legend.position = 'bottom')
pfas_sent_plot3
```



Source: Nexis Uni