assignment_2_sentiment_analysis1

Marie Rivers

4/14/2022

Sentiment Analysis I

This assignment uses sentiment analysis to... xxx

```
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'
```

```
## Creating LNToutput from 1 file...

##  ...files loaded [0.18 secs]

##  ...articles split [0.21 secs]

##  ...lengths extracted [0.21 secs]

##  ...headlines extracted [0.21 secs]

##  ...newspapers extracted [0.21 secs]

##  ...dates extracted [0.23 secs]
```

```
##  ...authors extracted [0.23 secs]

##  ...sections extracted [0.23 secs]

##  ...editions extracted [0.23 secs]

##  ...dates converted [0.24 secs]

##  ...metadata extracted [0.24 secs]

##  ...article texts extracted [0.25 secs]

##  ...superfluous whitespace removed [0.27 secs]

## Elapsed time: 0.27 secs
```

```
# 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$
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

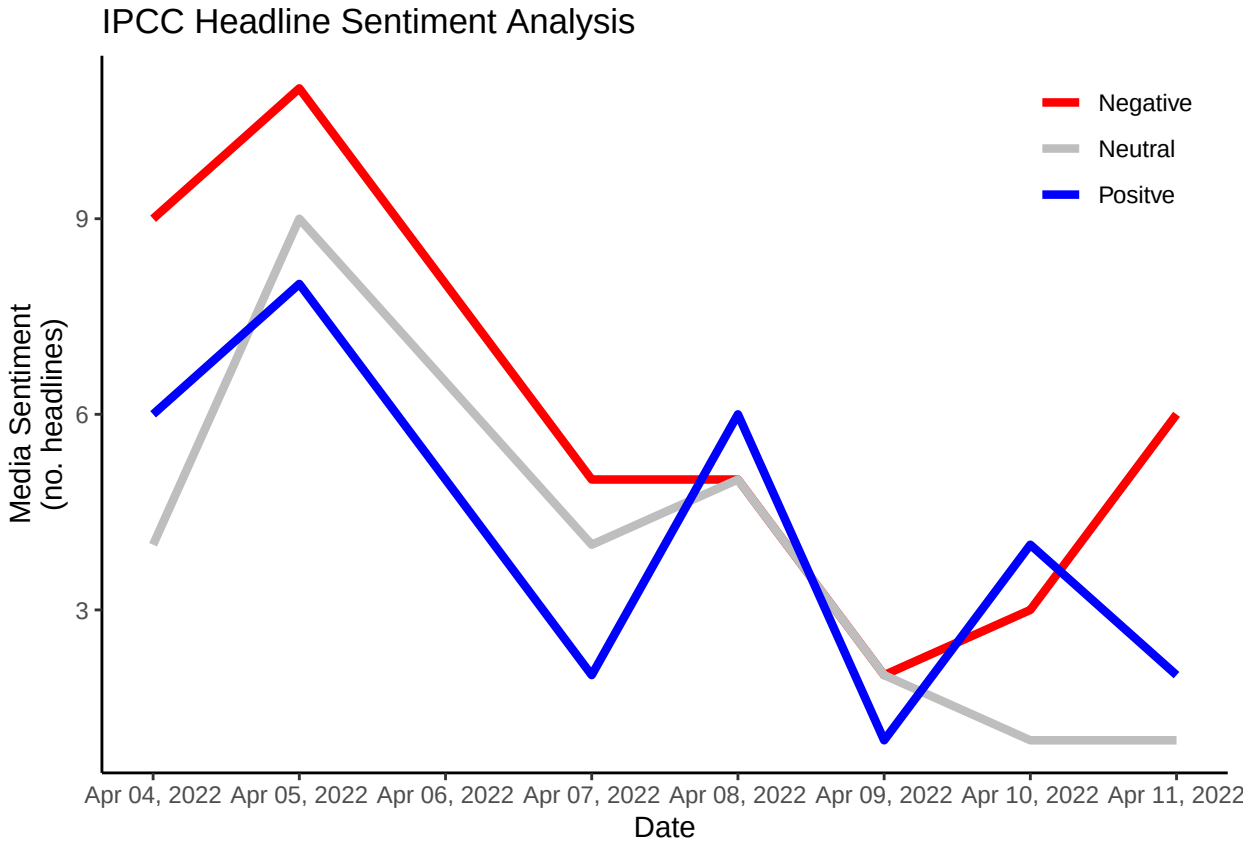
```
paragraphs_dat <- data_frame(element_id = paragraphs_df$Art_ID, Text  = paragraphs_df$Paragraph)
dat3 <- inner_join(dat2,paragraphs_dat, by = "element_id")
```

```
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")
```




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