Sentiment Analysis II

Paloma Cartwright

2022-04-25

Read in Twitter Data

```
raw_tweets <- read_csv(here("IPCC_tweets.csv"))
```

```
## New names:
## * `` -> ...1

## Rows: 2411 Columns: 84

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (33): Query Name, Date, Title, Snippet, Url, Domain, Sentiment, Emotion...
## dbl  (23): ...1, Query Id, Facebook Comments, Facebook Likes, Facebook Share...
## lgl  (27): Assignment, Category Details, Checked, Display URLs, Facebook Aut...
## time  (1): Time

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dat <- raw_tweets[,c(4,6)] # Extract Date and Title fields

tweets <- tibble(text = dat$Title,
                 id = seq(1:length(dat$Title)),
                 date = as.Date(dat$Date,'%m/%d/%y'))
```

1. Clean up Data

```
tweets$text <- gsub("http[^[:space:]]*", "",tweets$text)
tweets$text <- str_to_lower(tweets$text)
tweets$text <- gsub("@*", "", tweets$text)
```

Tokenize Data and Join sentiment words

```
bing_sent <- get_sentiments('bing')
nrc_sent <- get_sentiments('nrc')

words <- tweets %>%
  select(id, date, text) %>%
  unnest_tokens(output = word, input = text, token = "words") %>%
  anti_join(stop_words, by = "word") %>%
  left_join(bing_sent, by = "word") %>%
  left_join(
    tribble(
      ~sentiment, ~sent_score,
      "positive", 1,
      "negative", -1),
    by = "sentiment")
```

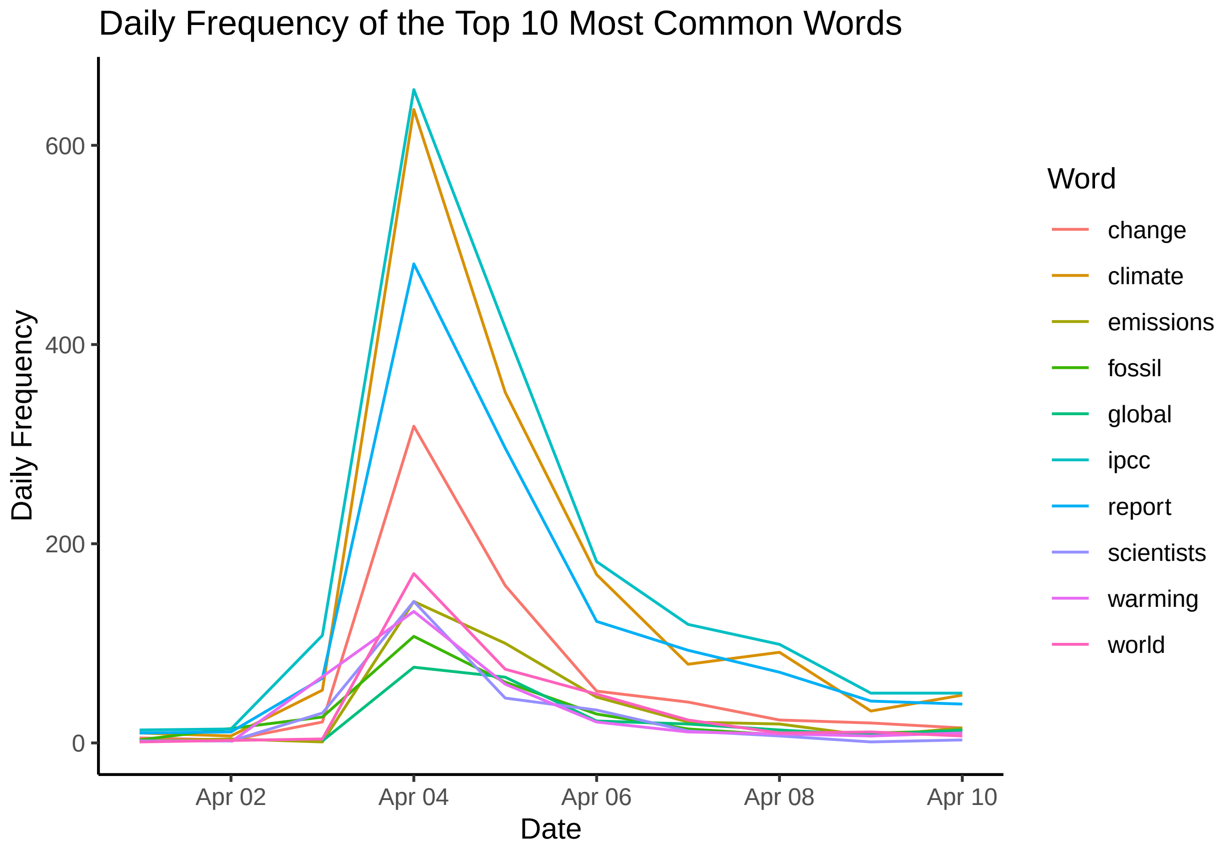
2. Compare the ten most common terms in the tweets per day

```
top_ten <- words %>%
  group_by(word) %>%
  summarize(count = n()) %>%
  ungroup() %>%
  slice_max(order_by = count, n = 10)

common <- words %>%
  filter(word %in% as.list(top_ten$word)) %>%
  select(date, word) %>%
  group_by(date, word) %>%
  summarize(count = n())
```

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

```
ggplot(common, aes(x = date, y = count, color = as.factor(word))) +
  geom_line() +
  theme_classic() +
  labs(x = "Date",
       y = "Daily Frequency",
       title = "Daily Frequency of the Top 10 Most Common Words",
       color = "Word")
```

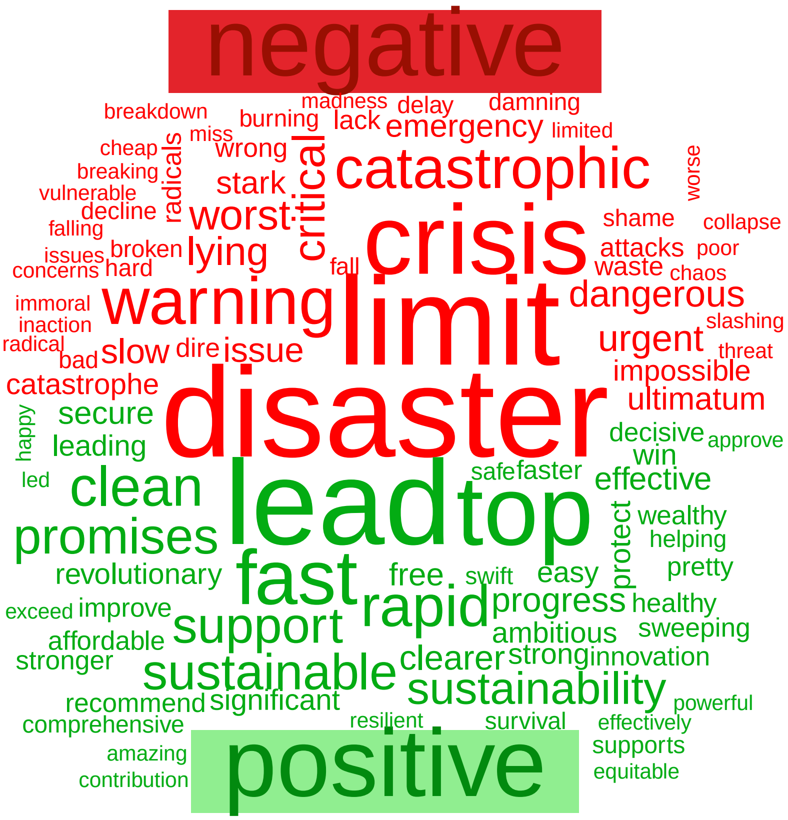


I noticed that both "IPCC" and "Climate" peaked on the day the IPCC released their report, which means that a lot of accounts were tweeting about it on April 4th and in the days after, even though on a lesser intensity.

3. Adjust the wordcloud in the "wordcloud" chunk by coloring the positive and negative words so they are identifiable.

```
words %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("red", "#03AC13"),
                   max.words = 100,
                   title.colors = c("#990F02", "#028A0F"),
                   title.bg.colors = c("#E3242B", "lightgreen"))
```

```
## Joining, by = c("word", "sentiment")
```



4. Let's say we are interested in the most prominent entities in the Twitter discussion. Which are the top 10 most tagged accounts in the data set?

```
dat2 <- raw_tweets[,c(4,6)] # Extract Date and Title fields

corpus <- corpus(dat2$Title) #enter quanteda

tokens <- tokens(corpus) #tokenize the text so each page is a list of tokens
tokens <- tokens(tokens,
                 remove_punct = TRUE,
                 remove_numbers = TRUE)

tokens <- tokens_select(tokens,
                        stopwords('english'),
                        selection='remove')
tokens <- tokens_tolower(tokens)

mention_tweets <- tokens(corpus, remove_punct = TRUE) %>%
  tokens_keep(pattern = "@*")

dfm_mention <- dfm(mention_tweets)

tstat_freq <- textstat_frequency(dfm_mention, n = 10)

tstat_freq <- data.frame(tstat_freq) %>%
  select(feature, frequency)
```

```
tstat_freq %>%
  kable(col.names = c("Account", "Frequency of Mentions")) %>%
  row_spec(0, bold = T)
```

Account	**Frequency of Mentions**
@ipcc_ch	131
@logicalindians	38
@antonioguterres	16
@nytimes	14
@yahoo	14
@potus	13
@un	12
@youtube	11
@conversationedu	10
@ipcc	9

The Twitter data download comes with a variable called "Sentiment" that must be calculated by Brandwatch. Use your own method to assign each tweet a polarity score (Positive, Negative, Neutral) and compare your classification to Brandwatch's (hint: you'll need to revisit the "raw_tweets" data frame).

```
tweets3 <- tibble(id = seq(1:length(dat$Title)),
                  date = as.Date(dat$Date,'%m/%d/%y'),
                  text = dat$Title)

tweets3$text <- gsub("http[^[:space:]]*", "",tweets3$text)
tweets3$text <- str_to_lower(tweets3$text)
tweets3$text <- gsub("@*", "", tweets3$text)

sent <- sentiment(tweets3$text) %>%
  group_by(element_id) %>%
  summarize(avg_sent = mean(sentiment))
```

```
## Warning: Each time `sentiment` is run it has to do sentence boundary disambiguation when a
## raw `character` vector is passed to `text.var`. This may be costly of time and
## memory.  It is highly recommended that the user first runs the raw `character`
## vector through the `get_sentences` function.
```

```
tweets_sent <- inner_join(tweets3, sent, by = c("id" = "element_id")) %>%
  mutate(polarity = case_when(
    avg_sent < 0 ~ "negative",
    avg_sent == 0 ~ "neutral",
    avg_sent > 0 ~ "positive"
  ))


my_sent <- tweets_sent %>%
  group_by(polarity) %>%
  summarize(count = n())

their_sent <- raw_tweets %>%
  group_by(Sentiment) %>%
  summarize(count = n())

their_pos <- their_sent$count[3]
their_neu <- their_sent$count[2]
their_neg <- their_sent$count[1]

my_pos <- my_sent$count[3]
my_neu <- my_sent$count[2]
my_neg <- my_sent$count[1]
```

When comparing the sentiment based on the method I used versus that Brandwatch calculated, there are very big differences. Their total positive tweets are 19 which is 1225 less than my positive sentiment. The majority of their tweets were scored neutral which differs from my majority positive tweets so there is clearly some differences. My sentiment scored the tweets more negative than Brandwatch did with 916 negative tweets compared to their 250 negative tweets.