03_sentiment_analysis_homework

Steven Cognac

4/13/2022

Import files

```
my_files <- list.files(pattern = ".docx",
                       path = here(),
                       full.names = TRUE,
                       recursive = TRUE,
                       ignore.case = TRUE)

# Object of class 'LNT output'
dat <- lnt_read(my_files)
```

```
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$Headline)
```

```
## 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.
```

```
# May be of use for assignment: using the full text from the articles
paragraphs_dat <- data_frame(element_id = paragraphs_df$Art_ID,
                             Text  = paragraphs_df$Paragraph)

dat3 <- inner_join(dat2, paragraphs_dat, by = "element_id")
```

Need to clean data above before chunk below

```
# can we create a similar graph to Figure 3A from Froelich et al.?
mytext <- get_sentences(dat3$Text)

sent <- sentiment(mytext)
```

```
sent_df <- inner_join(dat3, sent, by = "element_id")

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

```
## Warning: Each time ‘sentiment_by‘ 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.
```

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

```
## # A tibble: 8,903 x 7
##    element_id Date       Headline          Text  sentence_id word_count sentiment
##         <int> <date>     <chr>             <chr>       <int>      <int>     <dbl>
##  1         96 2022-04-05 Weekly Climate ~ "Apr~           1         22    -0.552
##  2         96 2022-04-05 Weekly Climate ~ " Th~           1         22    -0.552
##  3         96 2022-04-05 Weekly Climate ~ "Quo~           1         22    -0.552
##  4         96 2022-04-05 Weekly Climate ~ "Num~           1         22    -0.552
##  5         96 2022-04-05 Weekly Climate ~ "THI~           1         22    -0.552
##  6         96 2022-04-05 Weekly Climate ~ "By ~           1         22    -0.552
##  7         96 2022-04-05 Weekly Climate ~ "Sco~           1         22    -0.552
##  8         96 2022-04-05 Weekly Climate ~ "The~           1         22    -0.552
##  9         96 2022-04-05 Weekly Climate ~ "Jen~           1         22    -0.552
## 10         96 2022-04-05 Weekly Climate ~ "Que~           1         22    -0.552
## # ... with 8,893 more rows
```

```
sent_df$polarity <- ifelse(sent_df$sentiment <0, -1, ifelse(sent_df$sentiment > 0, 1, 0))
```

```
# unnest to word-level tokens, remove stop words, and join sentiment words
text_words2 <- dat3  %>%
  unnest_tokens(output = word, input = Text, token = 'words')

 # break text into individual words
sent_words <- text_words2 %>%

  # returns only the rows without stop words
  anti_join(stop_words, by = 'word') %>%

  # joins and retains only sentiment words
  inner_join(get_sentiments("nrc"), by = 'word') %>%

  # fitler out positive & negative
  filter(!sentiment %in% c('positive', 'negative'))
```

```
sent_pct <- sent_words %>%
  group_by(Date, sentiment) %>%
  count(sentiment) %>%
  ungroup() %>%
  group_by(Date) %>%
  mutate(n_max_day = sum(n),
         percent = round((n/n_max_day)*100, 2))
```

```
ggplot(data = sent_pct) +
  geom_line(aes(x = Date, y = percent, fill = sentiment, color = sentiment)) +
  labs(title = "Percent of Emotion from",
       subtitle = "term = 'storm surge protection'",
       caption = "lexicon 'emotion' from Nexis-Uni") +
  theme_minimal()
```

```
## Warning: Ignoring unknown aesthetics: fill
```