DATA 607 Week 10 - Natural Language Processing

Kevin Kirby

2024-10-31

```
library(tidyverse)
library(tidytext)
library(janeaustenr)
library(dplyr)
library(stringr)
library(readr)
```

Assignment Overview

This is assignment 8 from week ten of the fall 2024 edition of DATA 607.The assignment is as stated below, lightly edited for length and clarity:

You should start by getting the primary example code from chapter 2 of Text Mining with R on sentiment anlysis working in an R Markdown document. You should provide a citation to this base code. You're then asked to extend the code in two ways:

- Work with a different corpus of your choosing, and
- Incorporate at least one additional sentiment lexicon (possibly from another R package that you've found through research)."

The primary code example comes from the tidy-text-mining Github repo (Robinson and Silge), which is the official repository for the above noted book. The relevant Rmd parts are as follows:

"As discussed above, there are a variety of methods and dictionaries that exist for evaluating the opinion or emotion in text. The tidytext package provides access to several sentiment lexicons. Three general-purpose lexicons are

- `AFINN` from Finn Årup Nielsen,
- `bing` from Bing Liu and collaborators, and
- `nrc` from Saif Mohammad and Peter Turney.

All three of these lexicons are based on unigrams, i.e., single words. These lexicons contain many English words and the words are assigned scores for positive/negative sentiment, and also possibly emotions like joy, anger, sadness, and so forth. The `nrc` lexicon categorizes words in a binary fashion ("yes"/"no") into categories of positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise, and trust. The `bing` lexicon categorizes words in a binary fashion into positive and negative categories. The `AFINN` lexicon assigns words with a score that runs between -5 and 5, with negative scores indicating negative sentiment and positive scores indicating positive sentiment.

The function `get_sentiments()` allows us to get specific sentiment lexicons with the appropriate measures for each one.

```
library(tidytext)
library(tidyverse)
library(janeaustenr)
library(dplyr)
```

```
library(stringr)
library(ggplot2)

afinn <- get_sentiments("afinn")

bing <- get_sentiments("bing")

nrc <- get_sentiments("nrc")

# Prepare the text data in tidy format
tidy_books <- austen_books() %>%
  group_by(book) %>%
  mutate(
    linenumber = row_number(),
    chapter = cumsum(str_detect(text, regex("^chapter [\\\\divxlc]", ignore_case = TRUE)))
  ) %>%
  ungroup() %>%
  unnest_tokens(word, text)

# Join with NRC lexicon for sentiment "joy" in "Emma"
nrc_joy <- get_sentiments("nrc") %>%
  filter(sentiment == "joy")

tidy_books %>%
  filter(book == "Emma") %>%
  inner_join(nrc_joy) %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 301 x 2
##    word          n
##    <chr>     <int>
##  1 good        359
##  2 friend      166
##  3 hope        143
##  4 happy       125
##  5 love        117
##  6 deal         92
##  7 found        92
##  8 present      89
##  9 kind         82
## 10 happiness    76
## # i 291 more rows
```

```
nrc_joy <- nrc %>%
  filter(sentiment == "joy")

tidy_books %>%
  filter(book == "Emma") %>%
  inner_join(nrc_joy) %>%
  count(word, sort = TRUE)
```

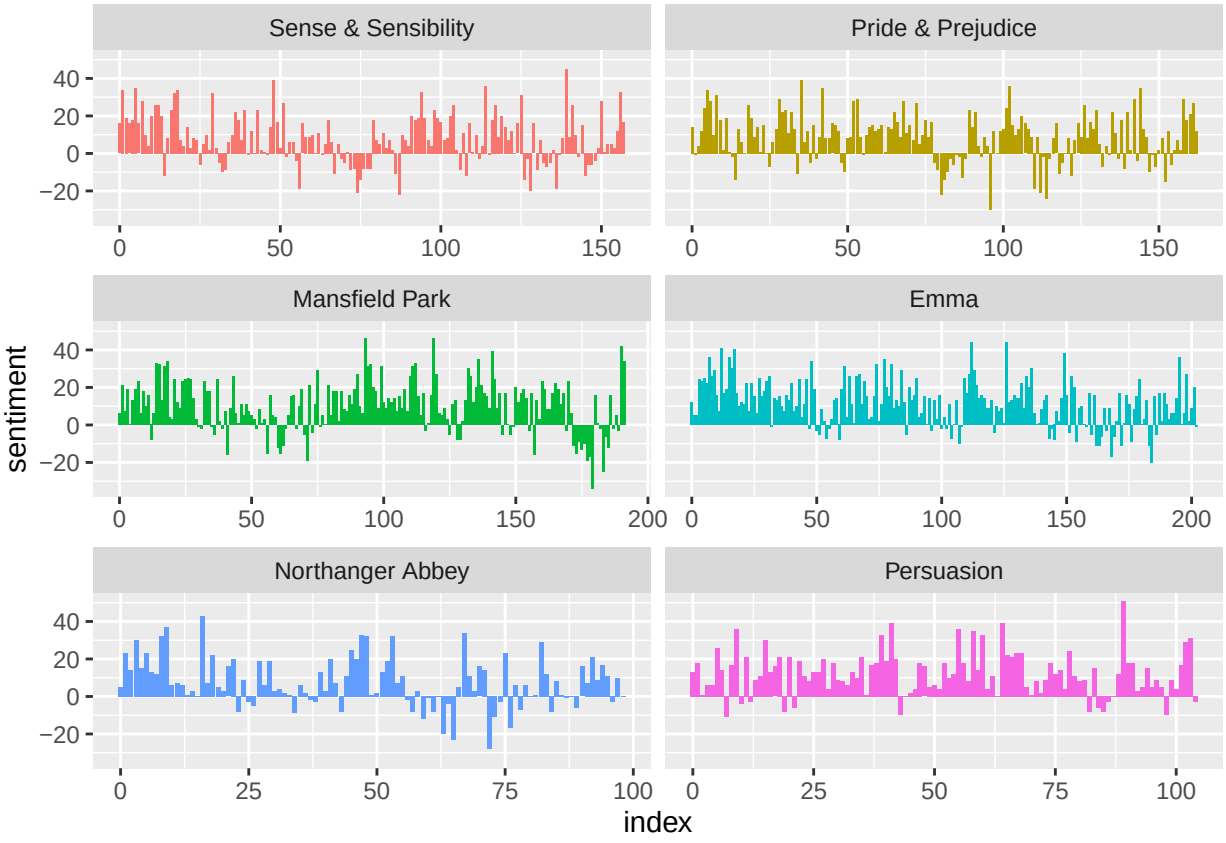
```
## # A tibble: 301 x 2
##    word          n
##    <chr>     <int>
```

```
##  1 good         359
##  2 friend       166
##  3 hope         143
##  4 happy        125
##  5 love         117
##  6 deal          92
##  7 found         92
##  8 present       89
##  9 kind          82
## 10 happiness     76
## # i 291 more rows
```

```
jane_austen_sentiment <- tidy_books %>%
  inner_join(get_sentiments("bing")) %>%
  count(book, index = linenumber %/% 80, sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) %>%
  mutate(sentiment = positive - negative)
```

```
## Warning in inner_join(., get_sentiments("bing")): Detected an unexpected many-to-many relationship be
## i Row 435434 of `x` matches multiple rows in `y`.
## i Row 5051 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

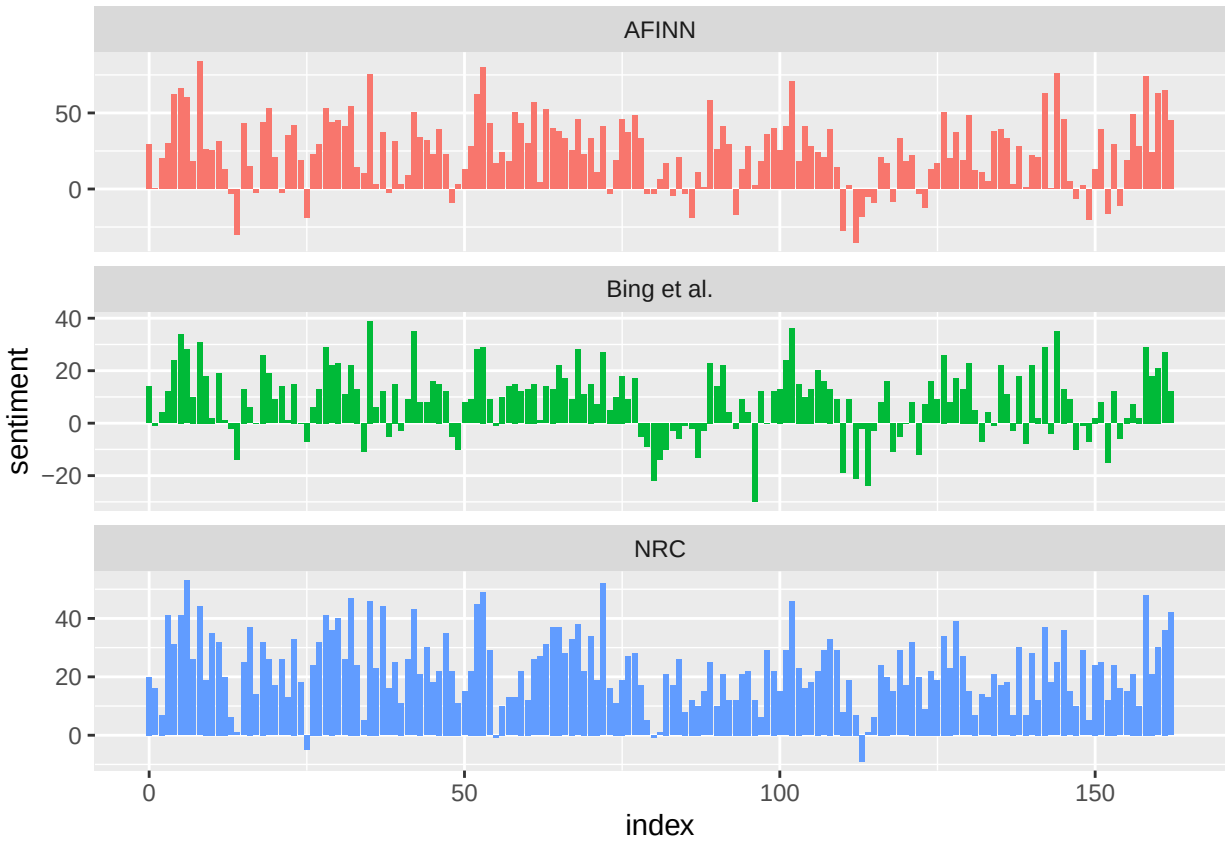
```
ggplot(jane_austen_sentiment, aes(index, sentiment, fill = book)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~book, ncol = 2, scales = "free_x")
```



```
pride_prejudice <- tidy_books %>%
  filter(book == "Pride & Prejudice")

afinn <- pride_prejudice %>%
  inner_join(afinn) %>%
  group_by(index = linenumber %/% 80) %>%
  summarise(sentiment = sum(value)) %>%
  mutate(method = "AFINN")

bing_and_nrc <- bind_rows(
  pride_prejudice %>%
    inner_join(get_sentiments("bing")) %>%
    mutate(method = "Bing et al."),
  pride_prejudice %>%
    inner_join(nrc %>%
                 filter(sentiment %in% c("positive",
                                         "negative"))
    ) %>%
    mutate(method = "NRC")) %>%
  count(method, index = linenumber %/% 80, sentiment) %>%
  pivot_wider(names_from = sentiment,
              values_from = n,
              values_fill = 0) %>%
  mutate(sentiment = positive - negative)
```

```
## Warning in inner_join(., nrc %>% filter(sentiment %in% c("positive", "negative"))): Detected an unexp
## i Row 215 of `x` matches multiple rows in `y`.
## i Row 5178 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
bind_rows(afinn,
          bing_and_nrc) %>%
  ggplot(aes(index, sentiment, fill = method)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~method, ncol = 1, scales = "free_y")
```




```
get_sentiments("nrc") %>%
  filter(sentiment %in% c("positive", "negative")) %>%
  count(sentiment)
```

```
## # A tibble: 2 x 2
##   sentiment     n
##   <chr>     <int>
## 1 negative   3316
## 2 positive   2308
```

```
nrc %>%
  filter(sentiment %in% c("positive", "negative")) %>%
  count(sentiment)
```

```
## # A tibble: 2 x 2
##   sentiment     n
##   <chr>     <int>
## 1 negative   3316
## 2 positive   2308
```

```
get_sentiments("bing") %>%
  count(sentiment)
```

```
## # A tibble: 2 x 2
##   sentiment     n
##   <chr>     <int>
## 1 negative   4781
## 2 positive   2005
```

” ## Overview of chosen data and analysis

I've chosen the App Store Reviews for a Mobile App from Kaggle, which contains the app store reviews for a mobile app, broken out by platform, contry, and device. I exported the data and then uploaded it to my GCP instance for import below.

For my additional Lexicon, I have chosen the Loughran-McDonald Lexicon. According to the University of Notre Dame's Software Repository for Accounting and Finance overview:

"The dictionary reports counts, proportion of total, average proportion per document, standard deviation of proportion per document, document count (i.e., number of documents containing at least one occurrence of the word), seven sentiment category identifiers, complexity, number of syllables, and source for each word (source is either 12of12inf or the year in which the word was added).

The sentiment categories are: negative, positive, uncertainty, litigious, strong modal, weak modal, and constraining."(Loughran and McDonald)

```
apps_url = 'https://storage.googleapis.com/data_science_masters_files/2024_fall/data_607_data_managemen

app_reviews_df <- read_delim(apps_url, delim = ";")

loughran_mc_lex <- get_sentiments("loughran")

head(app_reviews_df)
```

```
## # A tibble: 6 x 10
##   date       platform country   review       star user_id issue_flag likes_count
##   <chr>      <chr>    <chr>     <chr>       <dbl> <chr>   <chr>            <dbl>
## 1 7.07.2023  iOS      Australia Love the i~     5 13c954~ No                   1
## 2 12.08.2023 Android  India     The premiu~     4 945725~ No                   2
## 3 12.09.2023 iOS      UK        I can't sh~     5 e3d956~ No                   5
## 4 12.07.2023 Android  Brazil    The price ~     3 1fa559~ No                   0
## 5 24.09.2023 iOS      India     Smooth boo~     3 679346~ No                   2
## 6 20.09.2023 Android  USA       Premium ac~     2 ea1c97~ No                   4
## # i 2 more variables: dislike_count <dbl>, label <lgl>
```

Code implementation

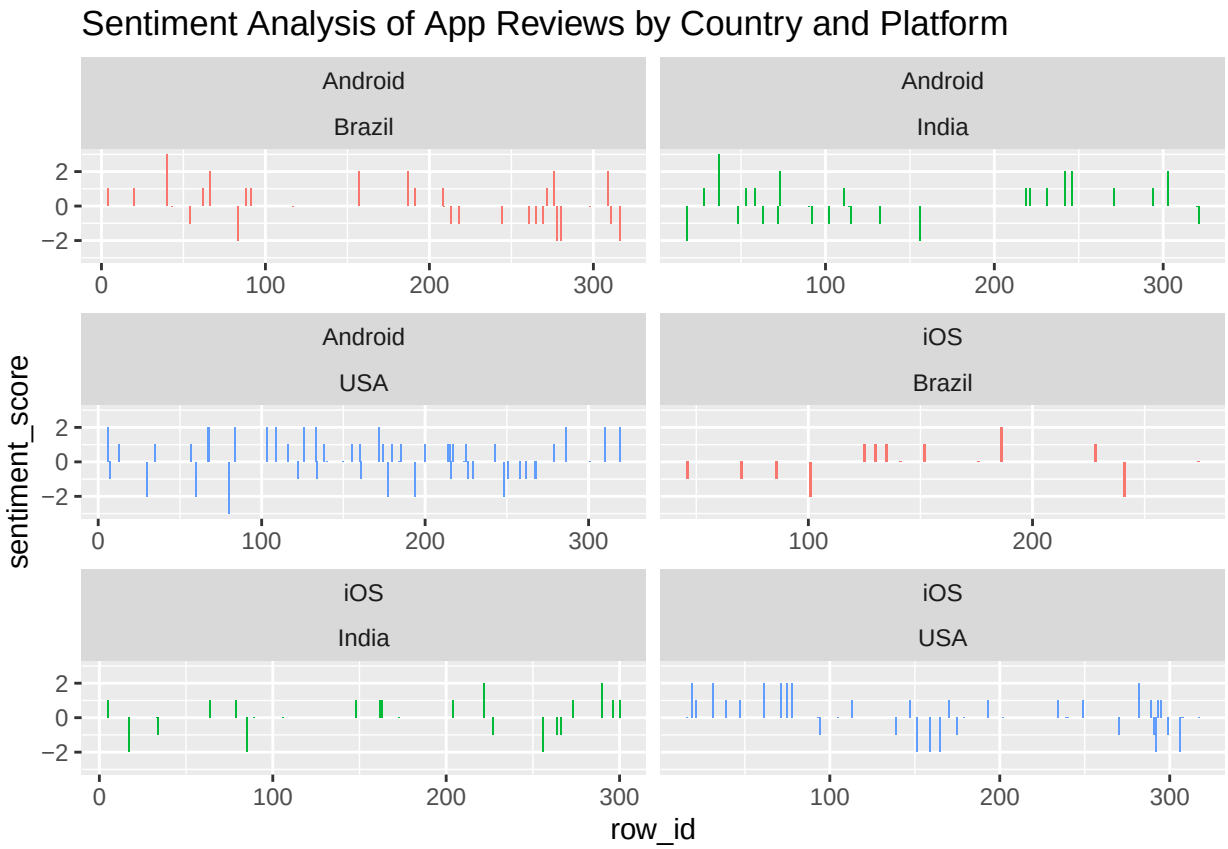
Here is ny implementation of the above code fronm the authors, with a component added that uses the Loughran method. This does the analysis by country, platform (operating system), and star reviews. Unsurprisingly, the higher the start reviews, the more positive the sentiment of the reviews. Not really any divergences by country or platform, either. It seems we are all, at the end of the day, remarkably similar in certain respects.

```
loughran_mc <- get_sentiments("loughran")

tidy_reviews_apps <- app_reviews_df %>%
  mutate(row_id = row_number()) %>%
  unnest_tokens(word, review)

nrc_joy_apps <- nrc %>%
  filter(sentiment == "joy")

tidy_reviews_apps %>%
  filter(country %in% c("USA", "India", "Brazil")) %>%
  inner_join(nrc_joy_apps,relationship = 'many-to-many') %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 18 x 2
```

```
##    word            n
##    <chr>       <int>
##  1 love           16
##  2 helpful         6
##  3 good            5
##  4 happy           4
##  5 holiday         4
##  6 inspire         4
##  7 share           4
##  8 perfect         3
##  9 finally         2
## 10 fun             2
## 11 improvement     2
## 12 pay             2
## 13 companion       1
## 14 deal            1
## 15 enjoy           1
## 16 excellent       1
## 17 found           1
## 18 pretty          1
```

```r
reviews_sentiment_apps <- tidy_reviews_apps %>%
  filter(country %in% c("USA", "India", "Brazil"), platform %in% c("iOS", "Android")) %>%
  inner_join(bing, relationship = 'many-to-many') %>%
  count(country, platform, row_id, sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) %>%
  mutate(sentiment_score = positive - negative)

ggplot(reviews_sentiment_apps, aes(row_id, sentiment_score, fill = country)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ platform + country, ncol = 2, scales = "free_x") +
  labs(title = "Sentiment Analysis of App Reviews by Country and Platform")
```

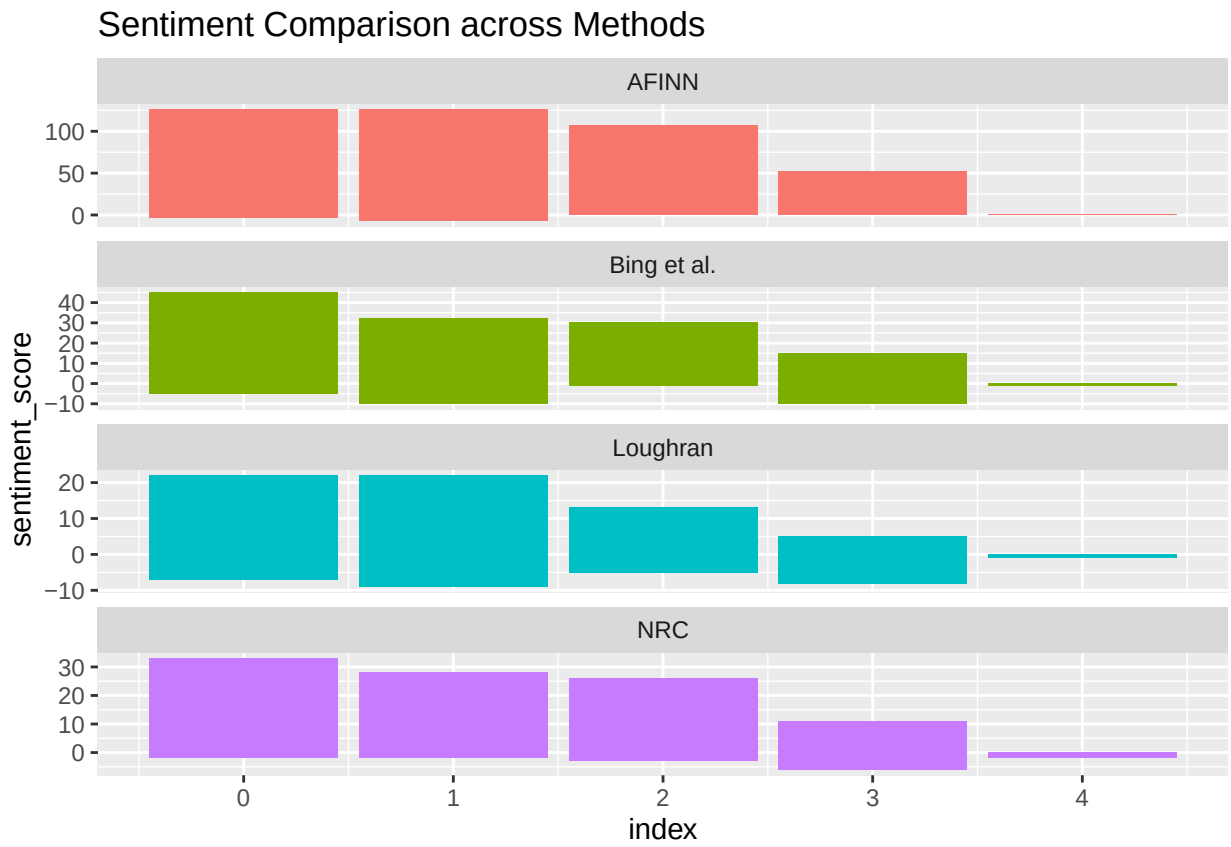



```
afinn_apps <- tidy_reviews_apps %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(country, platform, index = row_id %/% 80) %>%
  summarise(sentiment_score = sum(value)) %>%
  mutate(method = "AFINN")

bing_and_nrc_apps <- bind_rows(
  tidy_reviews_apps %>%
    inner_join(bing, relationship = "many-to-many") %>%
    mutate(method = "Bing et al."),
  tidy_reviews_apps %>%
    inner_join(nrc %>% filter(sentiment %in% c("positive", "negative"))) %>%
    mutate(method = "NRC")
) %>%
  count(method, country, platform, index = row_id %/% 80, sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) %>%
  mutate(sentiment_score = positive - negative)
```

```
## Warning in inner_join(., nrc %>% filter(sentiment %in% c("positive", "negative"))): Detected an unexp
## i Row 2714 of `x` matches multiple rows in `y`.
## i Row 1992 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
loughran_sentiment_apps <- tidy_reviews_apps %>%
  inner_join(loughran_mc %>% filter(sentiment %in% c("positive", "negative"))) %>%
  count(country, platform, index = row_id %/% 80, sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) %>%
```

```r
  mutate(sentiment_score = positive - negative, method = "Loughran")

sentiment_combined_apps <- bind_rows(afinn_apps, bing_and_nrc_apps, loughran_sentiment_apps)

ggplot(sentiment_combined_apps, aes(index, sentiment_score, fill = method)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ method, ncol = 1, scales = "free_y") +
  labs(title = "Sentiment Comparison across Methods")
```



Loughran, Tim, and Bill McDonald. "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks." *Journal of Finance*, vol. 66, no. 1, 2011, pp. 35–65, http://ssrn.com/abstract=1331573.

Robinson, David, and Julia Silge. *Sentiment Analysis in r.* 2021, https://github.com/dgrtwo/tidy-text-mining/blob/master/02-sentiment-analysis.Rmd.