Wk 10: Sentiment Analysis

Dirk Hartog

2023-11-01

Sentiment Analysis: A process of analyzing digital text to determine if the emotional tone of the message is positive, negative, or neutral

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(tidytext)
library(textdata)
library(stringr)
library(janeaustenr)
library(syuzhet)
```

Base code from chapter 2 that we will be working with. This includes a collection of works by author Jane Austen

```
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)

glimpse(tidy_books)
```

```
## Rows: 725,055
```

```
## Columns: 4
## $ book       <fct> Sense & Sensibility, Sense & Sensibility, Sense & Sensibili~
## $ linenumber <int> 1, 1, 1, 3, 3, 3, 5, 10, 10, 13, 13, 13, 13, 13, 13, 13, 13~
## $ chapter    <int> 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ word       <chr> "sense", "and", "sensibility", "by", "jane", "austen", "181~
```

Reference

Silge J, Robinson D. Welcome to Text Mining with R: A Tidy Approach. August 1, 2017. Section 2.2. O'Reilly Media. URL.

Extend the code in two ways:

a. Work with a different corpus of your choosing: Reference to Sherlock Holmes Data

```
devtools::install_github("EmilHvitfeldt/sherlock")
```

```
## Skipping install of 'sherlock' from a github remote, the SHA1 (38584034) has not changed since last i
##   Use `force = TRUE` to force installation
```

```
library(sherlock)
ls("package:sherlock")
```

```
## [1] "holmes"
```

```
tidy_holmes <- holmes %>%
  group_by(book) %>%
  mutate(
    linenumber = row_number(),
    chapter = cumsum(str_detect(text,
                                regex("^chapter [\\divxlc]",
                                      ignore_case = TRUE)))) %>%
  ungroup() %>%
  unnest_tokens(word, text)

data(stop_words) # loads a data frame of stop words

tidy_holmes <- tidy_holmes %>%
  anti_join(stop_words) %>% filter(chapter == !0)
```

```
## Joining with `by = join_by(word)`
```

```
head(tidy_holmes)
```

```
## # A tibble: 6 x 4
##   book                linenumber chapter word
##   <chr>                    <int>   <int> <chr>
## 1 A Study In Scarlet          29       1 chapter
## 2 A Study In Scarlet          30       1 sherlock
```

```
## 3 A Study In Scarlet         30       1 holmes
## 4 A Study In Scarlet         32       1 1878
## 5 A Study In Scarlet         32       1 degree
## 6 A Study In Scarlet         32       1 doctor
```

```
tidy_holmes %>% distinct(book)
```

```
## # A tibble: 7 x 1
##   book
##   <chr>
## 1 A Study In Scarlet
## 2 The Sign of the Four
## 3 A Scandal in Bohemia
## 4 The Hound of the Baskervilles
## 5 The Valley Of Fear
## 6 The Adventure of Wisteria Lodge
## 7 The Adventure of the Red Circle
```

b. Incorporate at least one additional sentiment lexicon. Through researching on the internet I came across this Rpubs article that mentioned a package called "syuzhet" that had a lexicon called Jockers.

- It contains 10,738 words
- Aims to incorporate emotional shifts in text
- Classifications: polarity and intensity. scores range from -1 to +1 (continuous)

Here I chose to compare how those sentiment scores compared to one of the lexicons from chapter 2 of Text Mining with R: A Tidy Approach. In this case I used the nrc lexicon to compare it to the syuzhet lexicon in the visualization that follows. The nrc lexicon associates a sentiment to a word and I used only the positive and negative words to create a sentiment total

I wanted to create one data frame to use for visualizing the trend throughout the story lines.

Step 1: Created seperate data frames with each of the lexicons dictionary of words

```
#
syuzhet <- tibble(get_sentiment_dictionary(dictionary = "syuzhet", language = "English"))
nrc <- get_sentiments("nrc") %>%
  filter(sentiment %in% c("positive",
                          "negative"))
```

Step 2: I joined each lexicons dictionaries with the words in the tidy_holmes data frame and then combined the two. I followed a similar procedure as the one in the book to create columns of the total sentiment in 25 lines of the book.

```
nrc_holmes <- tidy_holmes %>% inner_join(nrc, relationship = "many-to-many") %>%
  count(book, index = linenumber %/% 25, sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) %>%
  mutate(sentiment = positive - negative, lexicon = "nrc")
```

```
## Joining with `by = join_by(word)`
```

```
syuzhet_holmes <- tidy_holmes %>% inner_join(syuzhet, relationship = "many-to-many") %>%
  mutate(index = linenumber %/% 25) %>%
  group_by(book, index) %>%
  summarise(sentiment = sum(value)) %>%
  mutate(lexicon = "syuzhet")
```

```
## Joining with `by = join_by(word)`
## `summarise()` has grouped output by 'book'. You can override using the
## `.groups` argument.
```

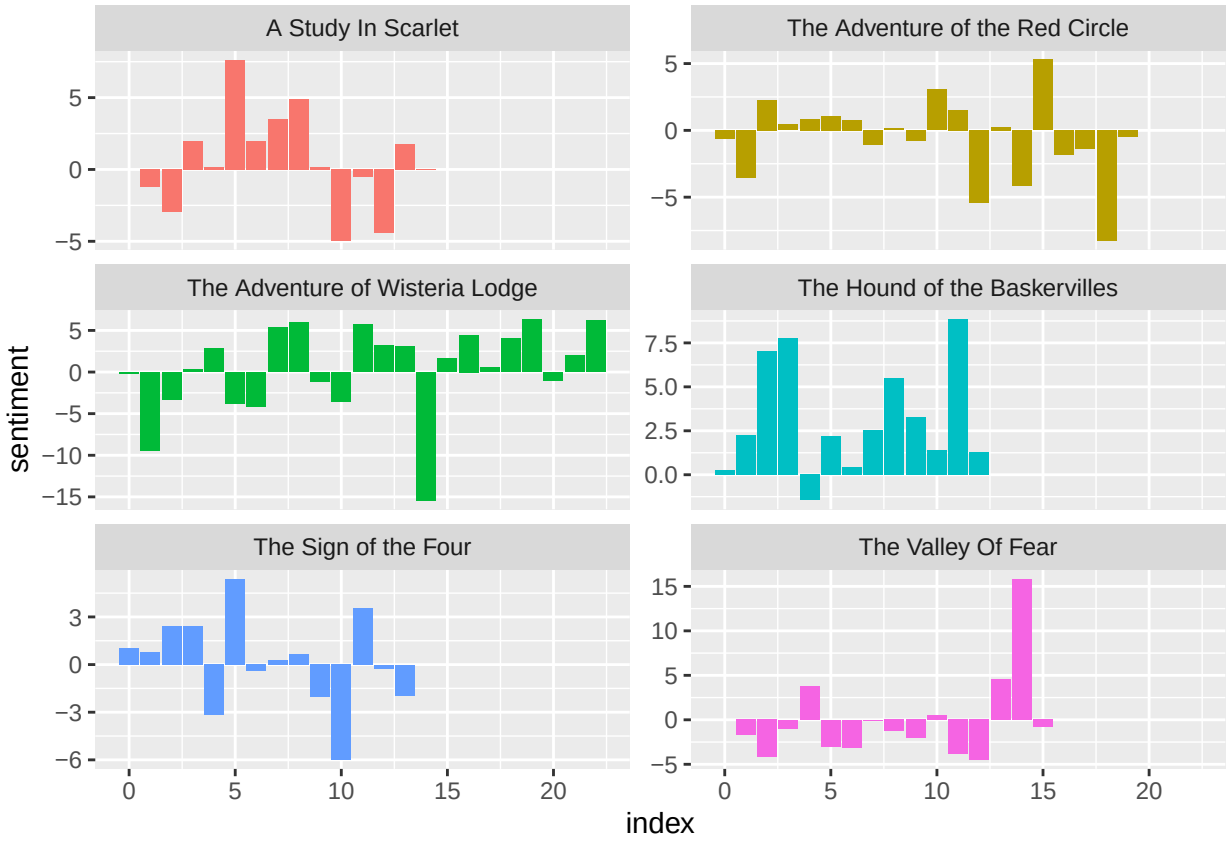
```
sentiment_df <- bind_rows(nrc_holmes, syuzhet_holmes) %>% select(c(1,2,5,6))

head(sentiment_df)
```

```
## # A tibble: 6 x 4
##   book                index sentiment lexicon
##   <chr>               <dbl>     <dbl> <chr>
## 1 A Study In Scarlet      1         0 nrc
## 2 A Study In Scarlet      2         0 nrc
## 3 A Study In Scarlet      3         6 nrc
## 4 A Study In Scarlet      4         6 nrc
## 5 A Study In Scarlet      5        14 nrc
## 6 A Study In Scarlet      6         6 nrc
```

Visualizations

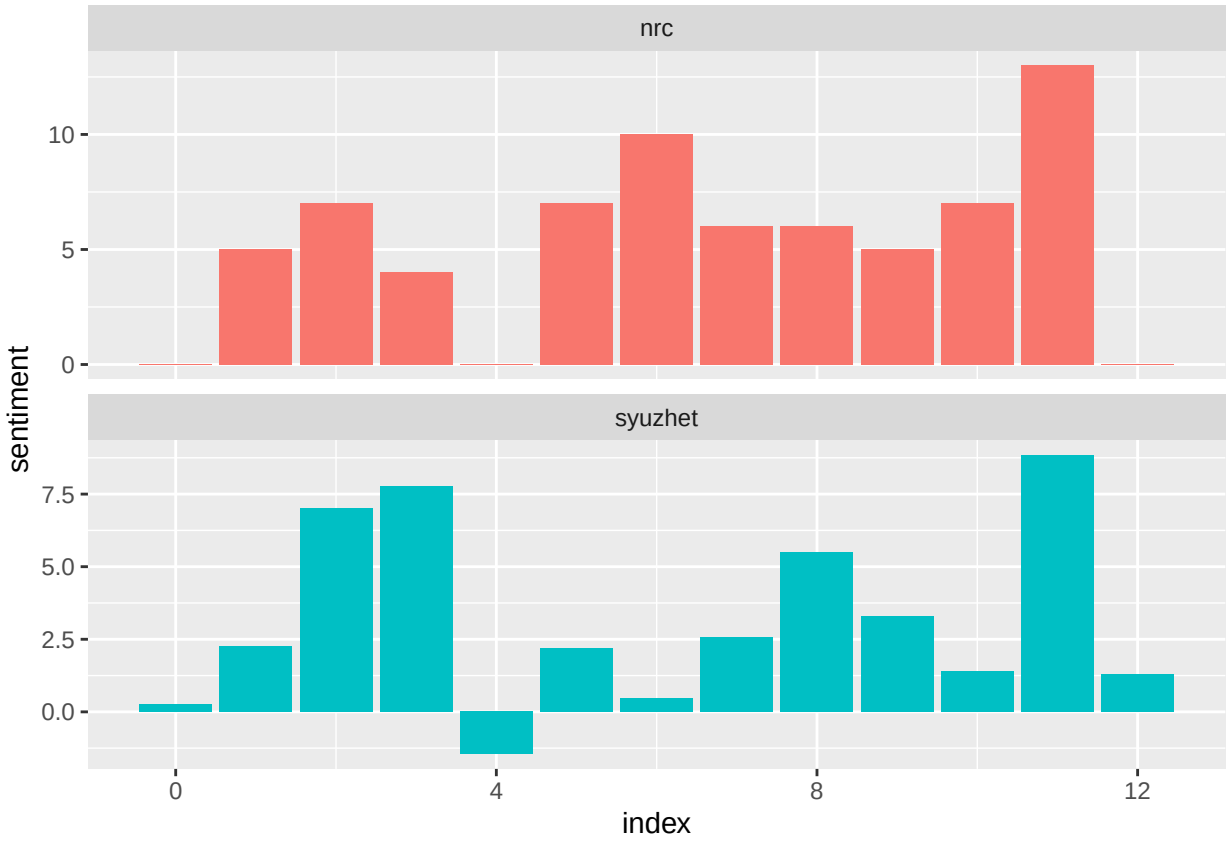
1. I thought it would be interesting to visualize the sentiment through the arc of the stories using the calcualted sentiment values from the syuzhet lexicon.

```
sentiment_df %>% filter(lexicon == "syuzhet") %>%
  ggplot(aes(index, sentiment, fill = book)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~book, ncol = 2, scales = "free_y")
```



2. Next I also wanted to compare sentiment through the story between two different lexicons

```
sentiment_df %>% filter(book == "The Hound of the Baskervilles") %>%
ggplot(aes(index, sentiment, fill = lexicon)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~lexicon, ncol = 1, scales = "free_y")
```



Conclusions

Across all the books in the Sherlock Holmes data set we see that the sentiment fluctuates from beginning to end and most books tend to end wit a neutral tone. When looking at the arc of "The Hound of the Baskervilles" each lexicon follows a similar trend except at index 6 were the sentiment looks to be evaluated quite differently between the two lexicons. Further investigation reveals that not all the same words were identified by each lexicon. This may have been from filtering out only the negative and positive words from the NRC dictionary.

```
bind_cols(tidy_holmes %>% mutate(index = linenumber %/% 25) %>%
  filter(index == 6 & book == "The Hound of the Baskervilles") %>%
  inner_join(syuzhet, relationship = "many-to-many") %>%
  select(SYUZHET_word = word, SYUZHET_value = value),

tidy_holmes %>% mutate(index = linenumber %/% 25) %>%
  filter(index == 6 & book == "The Hound of the Baskervilles") %>%
  inner_join(nrc, relationship = "many-to-many") %>%
  mutate(sentiment = ifelse(sentiment == "positive", 1, 0)) %>%
  select(NRC_word = word, NRC_value = sentiment)
)
```

```
## Joining with `by = join_by(word)`
## Joining with `by = join_by(word)`

## # A tibble: 18 x 4
```

```
##    SYUZHET_word  SYUZHET_value NRC_word     NRC_value
##    <chr>                 <dbl> <chr>            <dbl>
##  1 grave                 -0.5  grave                0
##  2 dear                   0.5  dear                 1
##  3 fellow                 0.4  fellow               1
##  4 amiable                0.5  amiable              1
##  5 absent                -0.6  absent               0
##  6 laughed                0.8  professional         1
##  7 incredulously         -0.4  career               1
##  8 wavering              -0.8  medical              1
##  9 smoke                 -0.25 medical              1
## 10 difficult             -0.5  visitor              1
## 11 professional           0.8  cross                0
## 12 visitor                0.25 winner               1
## 13 winner                 0.75 disease              0
## 14 prize                  0.8  author               1
## 15 entitled              -0.25 progress             1
## 16 disease               -1    march                1
## 17 freaks                -0.8  medical              1
## 18 progress               0.75 officer              1
```