textmining

Gemei

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✓ ggplot2 3.3.5 ✓ purrr 0.3.4  
## ✓ tibble 3.1.5 ✓ dplyr 1.0.7  
## ✓ tidyr 1.1.4 ✓ stringr 1.4.0  
## ✓ readr 2.0.2 ✓ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(tidytext)  
library(janeaustenr)  
library(stringr)  
library(gutenbergr)  
library(scales)

##   
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':  
##   
## discard

## The following object is masked from 'package:readr':  
##   
## col\_factor

##1. Pick three authors from Gutenberg.org and download their works. Compare the authors. Which two are more alike? Some suggestions: Mark Twain, Leo Tolstoy, Charles Dickens.

Twain <- gutenberg\_download(c(76,74,3179))

## Determining mirror for Project Gutenberg from http://www.gutenberg.org/robot/harvest

## Using mirror http://aleph.gutenberg.org

Tolstoy <- gutenberg\_download(c(1399,2142,2637))  
Dickens <- gutenberg\_download(c(98,766,1400))  
  
tidy\_Twain <- Twain %>%  
 unnest\_tokens(word, text) %>%  
 mutate(word = str\_extract(word, "[a-z']+")) %>% #remove the font/italics  
 anti\_join(stop\_words)

## Joining, by = "word"

tidy\_Twain %>%  
 count(word, sort = TRUE)

## # A tibble: 12,810 × 2  
## word n  
## <chr> <int>  
## 1 tom 1050  
## 2 time 671  
## 3 jim 374  
## 4 huck 349  
## 5 warn't 293  
## 6 de 270  
## 7 night 262  
## 8 day 250  
## 9 head 241  
## 10 en 240  
## # … with 12,800 more rows

tidy\_Tolstoy <- Tolstoy %>%  
 unnest\_tokens(word, text) %>%  
 mutate(word = str\_extract(word, "[a-z']+")) %>% #remove the font/italics  
 anti\_join(stop\_words)

## Joining, by = "word"

tidy\_Tolstoy %>%  
 count(word, sort = TRUE)

## # A tibble: 14,828 × 2  
## word n  
## <chr> <int>  
## 1 levin 1616  
## 2 vronsky 862  
## 3 anna 823  
## 4 time 744  
## 5 eyes 706  
## 6 kitty 673  
## 7 alexey 632  
## 8 love 626  
## 9 alexandrovitch 573  
## 10 life 562  
## # … with 14,818 more rows

tidy\_Dickens <- Dickens %>%  
 unnest\_tokens(word, text) %>%  
 mutate(word = str\_extract(word, "[a-z']+")) %>% #remove the font/italics  
 anti\_join(stop\_words)

## Joining, by = "word"

tidy\_Dickens %>%  
 count(word, sort = TRUE)

## # A tibble: 18,713 × 2  
## word n  
## <chr> <int>  
## 1 miss 1331  
## 2 time 1311  
## 3 hand 1030  
## 4 head 907  
## 5 dear 903  
## 6 looked 898  
## 7 night 809  
## 8 day 807  
## 9 aunt 778  
## 10 micawber 773  
## # … with 18,703 more rows

#frequency  
frequency\_by\_word\_across\_authors <- bind\_rows(mutate(tidy\_Twain, author = "Twain"),  
 mutate(tidy\_Tolstoy, author = "Tolstoy"),  
 mutate(tidy\_Dickens, author = "Dickens")) %>% mutate(word = str\_extract(word, "[a-z']+")) %>% count(author, word) %>%  
 group\_by(author) %>%  
 mutate(proportion = n / sum(n)) %>% select(-n) %>%  
 spread(author, proportion)  
  
frequency <- frequency\_by\_word\_across\_authors %>%   
 gather(author, proportion, `Tolstoy`:`Twain`)  
  
df\_Tolstoy <- frequency[frequency$author == "Tolstoy",]  
cor.test(data = df\_Tolstoy, ~ proportion + `Dickens`)

##   
## Pearson's product-moment correlation  
##   
## data: proportion and Dickens  
## t = 100.33, df = 9658, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.7044758 0.7240093  
## sample estimates:  
## cor   
## 0.7143817

df\_Twain <- frequency[frequency$author == "Twain",]  
cor.test(data = df\_Twain, ~ proportion + `Dickens`)

##   
## Pearson's product-moment correlation  
##   
## data: proportion and Dickens  
## t = 70.022, df = 8180, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.5984522 0.6255528  
## sample estimates:  
## cor   
## 0.6121823

So Tolstoy and Dickens are more alike.

##Let’s look at how the sentiment changes across the length of a book by looking at 80 lines at a time. Compare how sentiment changes in Victor Hugo’s Les Miserables and Charles Dickens’ A Tale of Two Cities. Look at negative vs positive sentiment. Then pick a sentiment like joy or anger or fear or … and see how that sentiment compares.

Hugo <- gutenberg\_download(135)  
Tale <- gutenberg\_download(98)  
tidy\_Hugo <- Hugo %>%  
 unnest\_tokens(word, text) %>%  
 mutate(word = str\_extract(word, "[a-z']+")) %>% #remove the font/italics  
 anti\_join(stop\_words)

## Joining, by = "word"

tidy\_Hugo %>%  
 count(word, sort = TRUE)

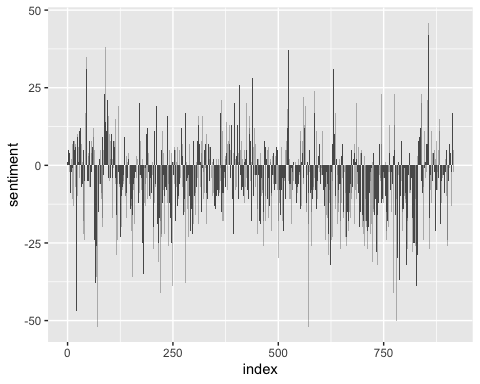
## # A tibble: 21,811 × 2  
## word n  
## <chr> <int>  
## 1 marius 1374  
## 2 jean 1238  
## 3 valjean 1120  
## 4 cosette 1023  
## 5 day 795  
## 6 time 762  
## 7 <NA> 757  
## 8 chapter 744  
## 9 rue 667  
## 10 de 642  
## # … with 21,801 more rows

tidy\_Hugo <- Hugo %>%   
 mutate(linenumber = row\_number(),  
 chapter = cumsum(str\_detect(text,  
 regex("^chapter [\\divxlc]",  
 ignore\_case = TRUE)))) %>%  
 ungroup()%>%  
 # use word so the inner\_join will match with the nrc lexicon  
 unnest\_tokens(word, text)  
  
Hugosentiment <- tidy\_Hugo %>%   
 inner\_join(get\_sentiments("bing")) %>%  
 count(index = linenumber %/% 80, sentiment) %>%   
 spread(sentiment, n, fill = 0) %>%  
 mutate(sentiment = positive - negative)

## Joining, by = "word"

Hugosentiment %>%  
 ggplot(aes(index, sentiment)) +   
 geom\_col(show.legend = FALSE,scales = "free\_x")

## Warning: Ignoring unknown parameters: scales

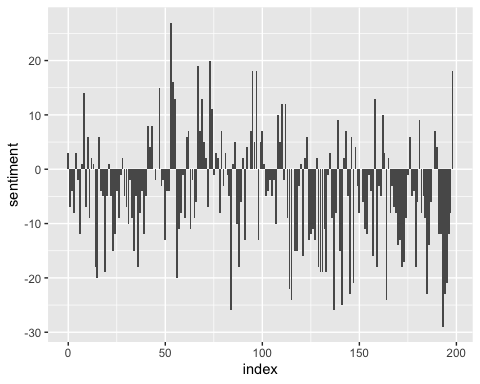


#a tale of two cities  
tidy\_Tale <- Tale %>%   
 mutate(linenumber = row\_number(),  
 chapter = cumsum(str\_detect(text,  
 regex("^chapter [\\divxlc]",  
 ignore\_case = TRUE)))) %>%  
 ungroup()%>%  
 # use word so the inner\_join will match with the nrc lexicon  
 unnest\_tokens(word, text)  
   
Talesentiment <- tidy\_Tale %>%   
 inner\_join(get\_sentiments("bing")) %>%  
 count(index = linenumber %/% 80, sentiment) %>%   
 spread(sentiment, n, fill = 0) %>%  
 mutate(sentiment = positive - negative)

## Joining, by = "word"

Talesentiment %>%  
 ggplot(aes(index, sentiment)) +   
 geom\_col(show.legend = FALSE,scales = "free\_x")

## Warning: Ignoring unknown parameters: scales



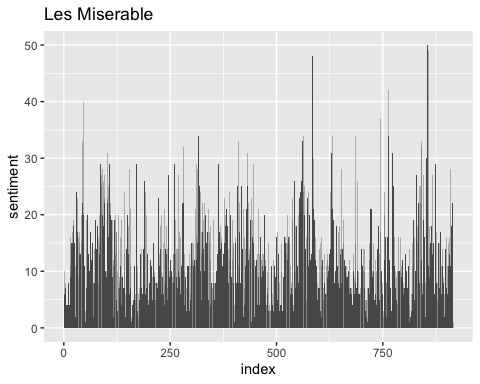
#specific sentiments  
joy=get\_sentiments("nrc")%>%filter(sentiment=="joy")  
  
#a tale of two cities  
Talesentiment <- tidy\_Tale %>%   
 inner\_join(joy) %>%  
 count(index = linenumber %/% 80, sentiment) %>%   
 spread(sentiment, n, fill = 0) %>%  
 mutate(sentiment = joy)

## Joining, by = "word"

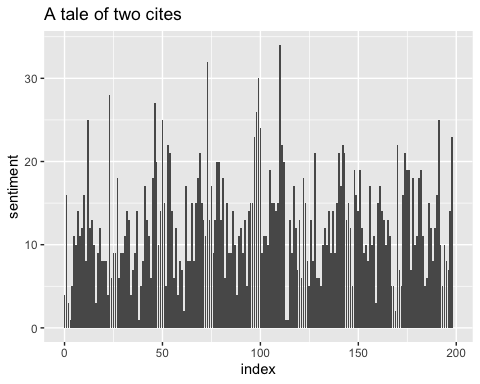
Hugosentiment <- tidy\_Hugo %>%   
 inner\_join(joy) %>%  
 count(index = linenumber %/% 80, sentiment) %>%   
 spread(sentiment, n, fill = 0) %>%  
 mutate(sentiment = joy)

## Joining, by = "word"

ggplot(Hugosentiment,aes(index,sentiment)) +geom\_col()+ggtitle("Les Miserable")



ggplot(Talesentiment,aes(index,sentiment)) +geom\_col()+ggtitle("A tale of two cites")



We can see that both of the books used a lot of negative words than positive words,in Les Miserable, the word joy is used more towards the end of the book, where in A tale of two cities, the word is used more in the first half of the book and less in the second half of the book.