Assignment Module 4 AG

Ani Gesheva

5/1/2021

## Description of project.

This is a sentiment analysis model - a model, which categorizes words based on their positive, negative or neutral sentiment, and on the magnitude of that sentiment. It’s a classification type of analysis.

The article defines sentiment analysis as the process of extracting opinions that have different positive, negative or neutral polarities. Sentiment analysis can reveal the nature of opinion that is reflected in documents, websites, social media feed, comments.

Hence the goal is to extract information about sentiment from the books.

The project will perform sentiment analysis using Inner Join.

It will be using the libraries ‘janeaustenr’, ‘stringr’ and ‘tidytext’.

The janeaustenr package provides text data - the text of Jane Austen’s books. Tidytext helps perform text analysis. I will convert the text of the books into a tidy format using the unnest\_tokens() function.

## Step 1: Installed the tidytext package and loaded it.

Tidytext package comprises of sentiment lexicons that are present in the dataset of ‘sentiments’.

install.packages(“tidytext”)

library(tidytext)

## Warning: package 'tidytext' was built under R version 4.0.5

sentiments

## # A tibble: 6,786 x 2  
## word sentiment  
## <chr> <chr>   
## 1 2-faces negative   
## 2 abnormal negative   
## 3 abolish negative   
## 4 abominable negative   
## 5 abominably negative   
## 6 abominate negative   
## 7 abomination negative   
## 8 abort negative   
## 9 aborted negative   
## 10 aborts negative   
## # ... with 6,776 more rows

Will make use of three general purpose lexicons: AFINN, bings and loughran.

These lexicons use unigrams - a type of n-gram models that consist of a sequence of 1 item, that is, a word collected from a given textual data.

The AFINN lexicon model scores the words in a range from -5 to 5. The increase in negativity corresponds to negative sentiment whereas an increase in positivity corresponds to positive one.

The bing lexicon model classifies the sentiment into a binary category of negative or positive.

The loughran model performs analysis of shareholder’s reports (I’d love to use that one some time to perform sentiment analysis on annual reports/10-Ks or letters to investors).

Will use the bing lexicons to extract the sentiments out of the data.

## Step 2: Retrieved the bing lexicons using the get\_sentiments() function.

get\_sentiments("bing")

## # A tibble: 6,786 x 2  
## word sentiment  
## <chr> <chr>   
## 1 2-faces negative   
## 2 abnormal negative   
## 3 abolish negative   
## 4 abominable negative   
## 5 abominably negative   
## 6 abominate negative   
## 7 abomination negative   
## 8 abort negative   
## 9 aborted negative   
## 10 aborts negative   
## # ... with 6,776 more rows

The bing lexicon consists of 6,786 words and their respective classifiction into positive or negative.

## Step 3: Loaded the following libraries: ‘janeaustenr’, ‘stringr’.

I loaded tidytext above in step 1.

In this project the data set is in the form of an R library/a package - it’s janeaustenr: Jane Austen’s Complete Novels.

It contains the full text for Jane Austen’s 6 completed novels “Sense and Sensibility”, “Pride and Prejudice”, “Mansfield Park”, “Emma”, “Northanger Abbey”, and “Persuasion”.

The code below imports Jane Austen library (text of her books), and imports stringr.

Then it assigns a data frame (object) called tidy\_data to the Jane Austen books text, groups it by book, and cleans the data.

Imported library dplyr as ungroup wouldn’t work w/o it.

The last line of code (the unnest\_tokens() fucntion) converts the text of Jane Austen’s books into a tidy format.

This step performs the tidy operation on the text such that each row contains a single word (cleaning/preparing data).

library(janeaustenr)

## Warning: package 'janeaustenr' was built under R version 4.0.5

library(stringr)

## Warning: package 'stringr' was built under R version 4.0.4

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.0.5

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

tidy\_data <- 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)

## Step 4: Used the bing lexicon and filter() over the words that correspond to joy/positive.

Used the book “Emma” and derived its words to implement the sentiment analysis model. (Note the original article cites “Sense and Sensibility” - but it’s “Emma” in the code below).

positive\_senti <- get\_sentiments("bing") %>%  
 filter(sentiment == "positive")  
tidy\_data %>%  
 filter(book == "Emma") %>%  
 semi\_join(positive\_senti) %>%  
 count(word, sort = TRUE)

## Joining, by = "word"

## # A tibble: 668 x 2  
## word n  
## <chr> <int>  
## 1 well 401  
## 2 good 359  
## 3 great 264  
## 4 like 200  
## 5 better 173  
## 6 enough 129  
## 7 happy 125  
## 8 love 117  
## 9 pleasure 115  
## 10 right 92  
## # ... with 658 more rows

There are 668 positive works in “Emma”. The most common is “well” (n=401), followed by “good” (n=359), “great” (n=264), “like”(n=200), and “better” (n=173).

## Step 5: Calculated sentiment.

Used the spread() function to segregate the data into separate columns of positive and negative sentiments. Then used the mutate() function to calculate the total sentiment - the difference between positive and negative sentiment.

library(tidyr)

## Warning: package 'tidyr' was built under R version 4.0.4

bing <- get\_sentiments("bing")  
Emma\_sentiment <- tidy\_data %>%  
 inner\_join(bing) %>%  
 count(book = "Emma" , index = linenumber %/% 80, sentiment) %>%  
 spread(sentiment, n, fill = 0) %>%  
 mutate(sentiment = positive - negative)

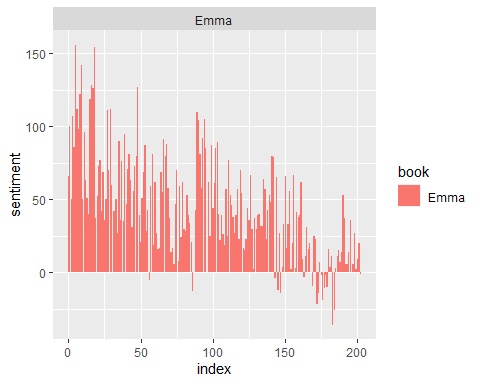
## Joining, by = "word"

## Step 6: Visualized the words in the book “Emma” based on their corresponding positive and negative scores.

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.0.5

ggplot(Emma\_sentiment, aes(index, sentiment, fill = book)) +  
 geom\_bar(stat = "identity", show.legend = TRUE) +  
 facet\_wrap(~book, ncol = 2, scales = "free\_x")



Based on the chart above it seems that “Emma” is a predominantly positive book.

## Step 7: Counted the most common positive and negative words present in the novel using Inner Join.

counting\_words <- tidy\_data %>%  
 inner\_join(bing) %>%  
 count(word, sentiment, sort = TRUE)

## Joining, by = "word"

head(counting\_words)

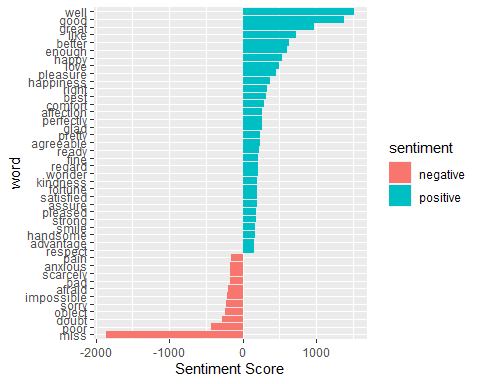
## # A tibble: 6 x 3  
## word sentiment n  
## <chr> <chr> <int>  
## 1 miss negative 1855  
## 2 well positive 1523  
## 3 good positive 1380  
## 4 great positive 981  
## 5 like positive 725  
## 6 better positive 639

Among the 2,585 entries (words), the most common are “miss”, which is a negative word (n=1,855), “well” (positive, n=1,523), “good” (positive, n=1,380), “great” (positive, n=981) and “like” (positive, n=725). This chart doesn’t entirely synch with above as we have the negative word “miss” as the most common.

## Step 8: Visualized the sentiment score.

Plot the scores along the axis that is labeled with both positive as well as negative words using the ggplot() function to visualize the data based on their scores.

counting\_words %>%  
 filter(n > 150) %>%  
 mutate(n = ifelse(sentiment == "negative", -n, n)) %>%  
 mutate(word = reorder(word, n)) %>%  
 ggplot(aes(word, n, fill = sentiment))+  
 geom\_col() +  
 coord\_flip() +  
 labs(y = "Sentiment Score")



## Step 9: Visualized with word cloud of the most recurring positive and negative words.

Used the comparision.cloud() function to plot both negative and positive words in a single word cloud.

install.packages(“reshape2”) install.packages(“wordcloud”)

library(reshape2)

## Warning: package 'reshape2' was built under R version 4.0.5

##   
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':  
##   
## smiths

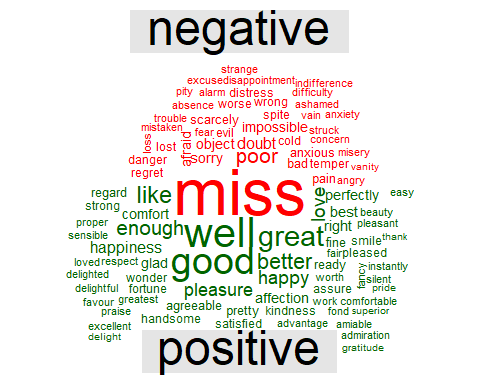
library(wordcloud)

## Warning: package 'wordcloud' was built under R version 4.0.5

## Loading required package: RColorBrewer

tidy\_data %>%  
 inner\_join(bing) %>%  
 count(word, sentiment, sort = TRUE) %>%  
 acast(word ~ sentiment, value.var = "n", fill = 0) %>%  
 comparison.cloud(colors = c("red", "dark green"),  
 max.words = 100)

## Joining, by = "word"



## Conclusion.

The last two visualizations confirm the previous points - that while the text and sentiment is mostly positive, there is one negative word (“miss”) that is most common and more common then the highest-frequency positive word (“well”).