# Data607 - Week 10 - Sentiment Analysis

Peter Gatica

04-18-2021

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidytext)
library(tinytex)
library(gutenbergr)
library(janeaustenr)
library(tidyverse)
library(RCurl)
```

```
##
## Attaching package: 'RCurl'

## The following object is masked from 'package:tidyr':
##
##     complete
```

```
library(knitr)
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

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

**Recreate and analyze primary code from textbook Welcome to Text Mining with R [@silge_robinson_text_mining_2017]. Recreating the code to analyze sentence sentimentality.**

@book{silge_robinson_text_mining_2017, author = {Julia Silge, David Robinson}, title = {Welcome to Text Mining with R}, publisher = {O“”’Reilly Media, Inc CA}, year = {2017}, isbn = {978-1491981658}, url = {https://github.com/dgrtwo/tidy-text-mining} }

**Recreating the code from Chapter 2 for sentence sentiment analysis** Some sentiment analysis algorithms look beyond only unigrams (i.e. single words) to try to understand the sentiment of a sentence as a whole. These algorithms try to understand that

```
I am not having a good day.
```

is a sad sentence, not a happy one, because of negation. R packages included coreNLP (T. Arnold and Tilton 2016), cleanNLP (T. B. Arnold 2016), and sentimentr (Rinker 2017) are examples of such sentiment analysis algorithms. For these, we may want to tokenize text into sentences, and it makes sense to use a new name for the output column in such a case.

```
(p_and_p_sentences <- tibble(text = prideprejudice) %>%
  unnest_tokens(sentence, text, token = "sentences"))  # unnest tokens into a field called sentence wit
```

```
## # A tibble: 15,545 x 1
##    sentence
##    <chr>
##  1 "pride and prejudice"
##  2 "by jane austen"
##  3 "chapter 1"
##  4 "it is a truth universally acknowledged, that a single man in possession"
##  5 "of a good fortune, must be in want of a wife."
##  6 "however little known the feelings or views of such a man may be on his"
##  7 "first entering a neighbourhood, this truth is so well fixed in the minds"
##  8 "of the surrounding families, that he is considered the rightful property"
##  9 "of some one or other of their daughters."
## 10 "\"my dear mr."
## # ... with 15,535 more rows
```

```
                                                        # sentences
```

```
(p_and_p_sentences$sentence[2])
```

```
## [1] "by jane austen"
```

The sentence tokenizing does seem to have a bit of trouble with UTF-8 encoded text, especially with sections of dialogue; it does much better with punctuation in ASCII. One possibility, if this is important, is to try using iconv(), with something like iconv(text, to = ‘latin1’) in a mutate statement before unnesting.

Another option in unnest_tokens() is to split into tokens using a regex pattern. We could use this, for example, to split the text of Jane Austen’s novels into a data frame by chapter.

```r
(austen_chapters <- austen_books() %>%  # pipe austen_books to group_by()
  group_by(book) %>%                    # group output by book
    unnest_tokens(chapter, text, token = "regex",     #  unnest tokens by chapters using regex to find
                pattern = "Chapter|CHAPTER [\\dIVXLC]") %>%  # chapters.  each row contains the all
  ungroup())                                                # sentences in a chapter
```

```
## # A tibble: 275 x 2
##    book              chapter
##    <fct>             <chr>
##  1 Sense & Sensibi~ "sense and sensibility\n\nby jane austen\n\n(1811)\n\n\n\n\~
##  2 Sense & Sensibi~ "\n\n\nthe family of dashwood had long been settled in suss~
##  3 Sense & Sensibi~ "\n\n\nmrs. john dashwood now installed herself mistress of~
##  4 Sense & Sensibi~ "\n\n\nmrs. dashwood remained at norland several months; no~
##  5 Sense & Sensibi~ "\n\n\n\"what a pity it is, elinor,\" said marianne, \"that~
##  6 Sense & Sensibi~ "\n\n\nno sooner was her answer dispatched, than mrs. dashw~
##  7 Sense & Sensibi~ "\n\n\nthe first part of their journey was performed in too~
##  8 Sense & Sensibi~ "\n\n\nbarton park was about half a mile from the cottage. ~
##  9 Sense & Sensibi~ "\n\n\nmrs. jennings was a widow with an ample jointure.  s~
## 10 Sense & Sensibi~ "\n\n\nthe dashwoods were now settled at barton with tolera~
## # ... with 265 more rows
```

```r
(austen_chapters %>%
  group_by(book) %>%
  summarise(chapters = n()))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 6 x 2
##   book                chapters
##   <fct>                  <int>
## 1 Sense & Sensibility       51
## 2 Pride & Prejudice         62
## 3 Mansfield Park            49
## 4 Emma                      56
## 5 Northanger Abbey          32
## 6 Persuasion                25
```

1. Let's get the list of negative words from the Bing lexicon.

```r
(bingnegative <- get_sentiments("bing") %>%
  filter(sentiment == "negative"))
```

```
## # A tibble: 4,781 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 4,771 more rows
```

```
(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))
```

```
## # A tibble: 725,055 x 4
##    book                linenumber chapter word
##    <fct>                    <int>   <int> <chr>
##  1 Sense & Sensibility          1       0 sense
##  2 Sense & Sensibility          1       0 and
##  3 Sense & Sensibility          1       0 sensibility
##  4 Sense & Sensibility          3       0 by
##  5 Sense & Sensibility          3       0 jane
##  6 Sense & Sensibility          3       0 austen
##  7 Sense & Sensibility          5       0 1811
##  8 Sense & Sensibility         10       1 chapter
##  9 Sense & Sensibility         10       1 1
## 10 Sense & Sensibility         13       1 the
## # ... with 725,045 more rows
```

2. Make a data frame of how many words are in each chapter so we can normalize for the length of chapters.

```
(wordcounts <- tidy_books %>%
  group_by(book, chapter) %>%
  summarize(words = n()))
```

```
## `summarise()` regrouping output by 'book' (override with `.groups` argument)
```

```
## # A tibble: 275 x 3
## # Groups:   book [6]
##    book                chapter words
##    <fct>                 <int> <int>
##  1 Sense & Sensibility       0     7
##  2 Sense & Sensibility       1  1571
##  3 Sense & Sensibility       2  1970
##  4 Sense & Sensibility       3  1538
##  5 Sense & Sensibility       4  1952
##  6 Sense & Sensibility       5  1030
##  7 Sense & Sensibility       6  1353
##  8 Sense & Sensibility       7  1288
```

```
##  9 Sense & Sensibility       8  1256
## 10 Sense & Sensibility       9  1863
## # ... with 265 more rows
```

3. Find the number of negative words in each chapter and divide by the total words in each chapter.

For each book, which chapter has the highest proportion of negative words?

```
(tidy_books %>%
  semi_join(bingnegative) %>%
  group_by(book, chapter) %>%
  summarize(negativewords = n()) %>%
  left_join(wordcounts, by = c("book", "chapter")) %>%
  mutate(ratio = negativewords/words) %>%
  filter(chapter != 0) %>%
  slice_max(ratio, n = 1) %>%
  ungroup())
```

```
## Joining, by = "word"
```

```
## `summarise()` regrouping output by 'book' (override with `.groups` argument)
```

```
## # A tibble: 6 x 5
##   book                chapter negativewords words  ratio
##   <fct>                 <int>         <int> <int>  <dbl>
## 1 Sense & Sensibility      43           161  3405 0.0473
## 2 Pride & Prejudice        34           111  2104 0.0528
## 3 Mansfield Park           46           173  3685 0.0469
## 4 Emma                     15           151  3340 0.0452
## 5 Northanger Abbey         21           149  2982 0.0500
## 6 Persuasion                4            62  1807 0.0343
```

### Import another lexicon (From twitter on airline sentiment)

Import bing sentiment words to use as a look up.

```
lookup_bing <- get_sentiments("bing")
```

Import the csv file airline review tweets as found on kaggle.com (https://www.kaggle.com/crowdflower/twitter-airline-sentiment)

```
# filename <- getURL("https://raw.githubusercontent.com/audiorunner13/Masters-Coursework/main/DATA607%2
# airline_tweets_src <- read.csv(text = filename,na.strings = "")
filename <- "/Users/Audiorunner13/CUNY MSDS Course Work/DATA607 Spring 2021/Week10/archive/Tweets.csv"
airline_tweets_src <- read.csv(filename)
```

```
airline_tweets_src$text <- tolower(airline_tweets_src$text) %>% str_replace("^@[a-z]* ", "")
```

```
head(airline_tweets <- airline_tweets_src %>% select(, airline, text, airline_sentiment),10)
```

```
##          airline
## 1  Virgin America
## 2  Virgin America
## 3  Virgin America
## 4  Virgin America
## 5  Virgin America
## 6  Virgin America
## 7  Virgin America
## 8  Virgin America
## 9  Virgin America
## 10 Virgin America
##
## 1
## 2                                                                  plus you’ve added commercials to t
## 3                                                                   i didn’t today... must mean i ne
## 4            it’s really aggressive to blast obnoxious "entertainment" in your guests’ faces &amp; th
## 5                                                                                   and it’s a really
## 6  seriously would pay $30 a flight for seats that didn’t have this playing.\nit’s really the only ba
## 7                                                            yes, nearly every time i fly vx this "ear
## 8                               really missed a prime opportunity for men without hats parody, there.
## 9                                                                                           well, i
## 10                                                          it was amazing, and arrived an hour early
##    airline_sentiment
## 1            neutral
## 2           positive
## 3            neutral
## 4           negative
## 5           negative
## 6           negative
## 7           positive
## 8            neutral
## 9           positive
## 10          positive
```

What are the most common joy words by airline? 1. We need to take the text of the review and convert the text to the tidy format using unnest_tokens(). 2. Also, set up some other columns to keep track of which line and text of the airline each word comes from 3. We use group_by and mutate to construct those columns.

```
(tidy_airline_reviews <- airline_tweets %>%
  group_by(airline) %>%
  mutate(
    review = row_number())  %>%
  ungroup() %>%
  unnest_tokens(word, text))
```

```
## # A tibble: 247,413 x 4
##    airline        airline_sentiment review word
##    <chr>          <chr>              <int> <chr>
##  1 Virgin America neutral                1 what
```

```
##  2 Virgin America neutral                1 dhepburn
##  3 Virgin America neutral                1 said
##  4 Virgin America positive               2 plus
##  5 Virgin America positive               2 you've
##  6 Virgin America positive               2 added
##  7 Virgin America positive               2 commercials
##  8 Virgin America positive               2 to
##  9 Virgin America positive               2 the
## 10 Virgin America positive               2 experience
## # ... with 247,403 more rows
```

Next, let's filter() the data frame with the text from the books for the words from Emma and then use inner_join() to perform the sentiment analysis. What are the most common joy words in Emma? Let's use count() from dplyr.

```
(tidy_airline_reviews %>%      # pipe tidy_books content to filter()
  filter(airline == "Virgin America") %>%     # filter on the book Emma
  inner_join(lookup_bing) %>%        #  inner_join() on nrc_joy
  count(word, sort = TRUE))      #  get a count of each joy word and sort in descending order
```

```
## Joining, by = "word"
```

```
## # A tibble: 207 x 2
##    word         n
##    <chr>    <int>
##  1 love        27
##  2 great       19
##  3 best        15
##  4 thank       15
##  5 like        14
##  6 awesome     11
##  7 cool        11
##  8 problems    11
##  9 elevate     10
## 10 amazing      8
## # ... with 197 more rows
```

Count up how many positive and negative words there are for each airline.

We define an index here to keep track of where we are in the narrative; this index (using integer division) counts up sections of 80 lines of text for a better estimate than smaller or larger sections.

Use pivot_wider() so that we have negative and positive sentiment in separate columns.

Calculate a net sentiment (positive - negative).

```
(twitter_airline_sentiment <- tidy_airline_reviews %>%
  inner_join(lookup_bing) %>%
  count(airline, sentiment))
```

```
## Joining, by = "word"
```

```
## # A tibble: 12 x 3
```

```
##    airline        sentiment     n
##    <chr>          <chr>     <int>
##  1 American       negative   1621
##  2 American       positive   1263
##  3 Delta          negative    891
##  4 Delta          positive   1180
##  5 Southwest      negative   1006
##  6 Southwest      positive   1353
##  7 United         negative   2573
##  8 United         positive   1805
##  9 US Airways     negative   2022
## 10 US Airways     positive   1220
## 11 Virgin America negative    175
## 12 Virgin America positive    294
```

Multiply the negative counts by -1 for use with ggplot2

```
x <- 1
while (x < 13){
  if (twitter_airline_sentiment$sentiment[x] == "negative"){
     twitter_airline_sentiment$n[x] = twitter_airline_sentiment$n[x] * -1
  }
  x <- x + 1
}
twitter_airline_sentiment
```

```
## # A tibble: 12 x 3
##    airline        sentiment     n
##    <chr>          <chr>     <dbl>
##  1 American       negative  -1621
##  2 American       positive   1263
##  3 Delta          negative   -891
##  4 Delta          positive   1180
##  5 Southwest      negative  -1006
##  6 Southwest      positive   1353
##  7 United         negative  -2573
##  8 United         positive   1805
##  9 US Airways     negative  -2022
## 10 US Airways     positive   1220
## 11 Virgin America negative   -175
## 12 Virgin America positive    294
```

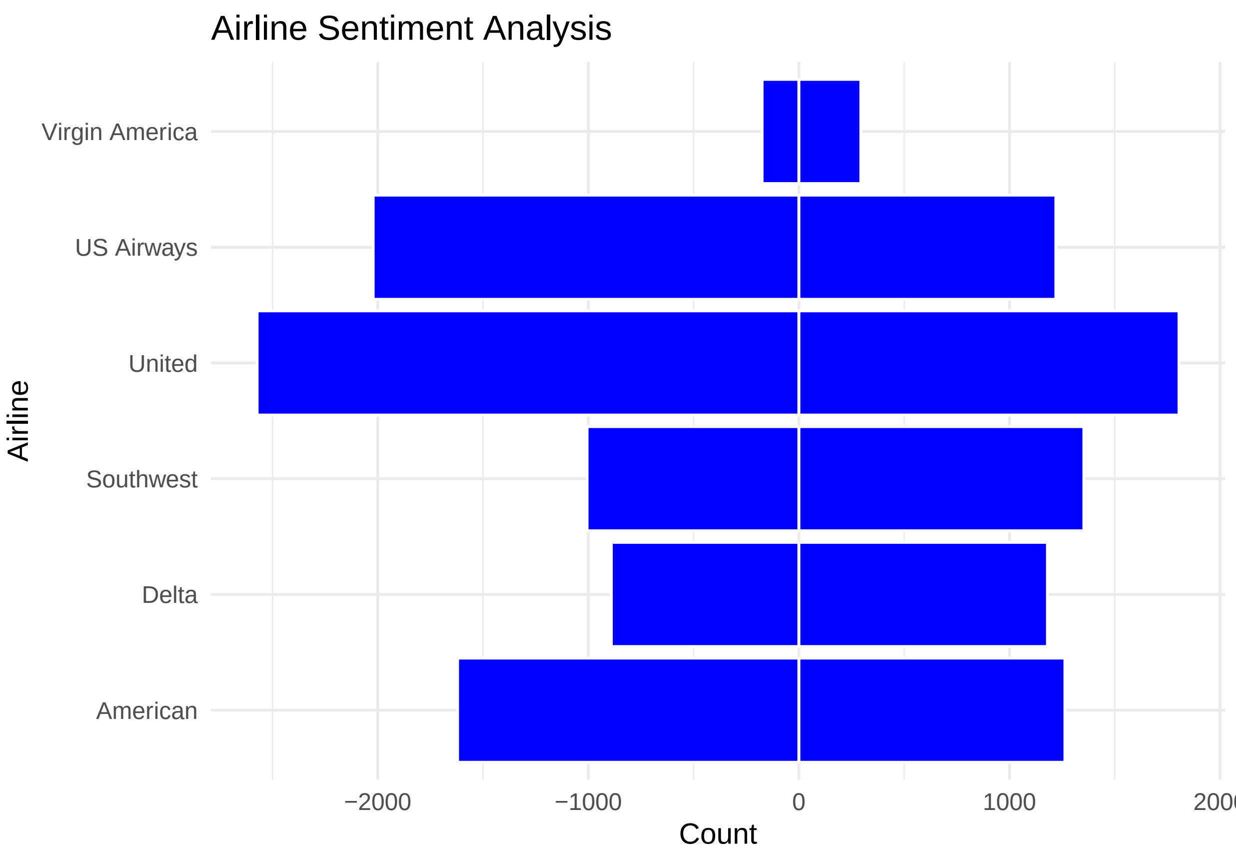
Rename columns for use with ggplot2

```
twitter_airline_sentiment <- twitter_airline_sentiment %>% rename(Airline = airline, Sentiment = sentim
```

Plot negative and postive counts by airline using ggplot2

```
ggplot(twitter_airline_sentiment, aes(x = Airline, y = Count)) +
  geom_bar(
    stat = "identity", position = position_stack(),
    color = "white", fill = "blue"
```

```
) +
labs(title = ("Airline Sentiment Analysis")) +
    theme_minimal() +
coord_flip()
```




As you can see from the plot that of the 6 major airlines United have the most negative reviews and US Airways has almost twice the negative reviews as positive. Southwest, Delta and US Virgin have more positive reviews than negative, however, Virgin America very few reviews compared to the other airlines.

```
(twitter_airline_sentiment %>%
pivot_wider(names_from = Sentiment, values_from = Count, values_fill = 0) %>%
  mutate(Sentiment = positive + negative) %>%
  rename(Negative = negative, Positive = positive))
```

```
## # A tibble: 6 x 4
##   Airline        Negative Positive Sentiment
##   <chr>             <dbl>    <dbl>     <dbl>
## 1 American          -1621     1263      -358
## 2 Delta              -891     1180       289
## 3 Southwest         -1006     1353       347
## 4 United            -2573     1805      -768
## 5 US Airways        -2022     1220      -802
## 6 Virgin America     -175      294       119
```

```
(airline_tweets %>%
  group_by(airline) %>%
```

```
  summarise(texts = n()) %>%
  ungroup)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 6 x 2
##   airline        texts
##   <chr>          <int>
## 1 American        2759
## 2 Delta           2222
## 3 Southwest       2420
## 4 United          3822
## 5 US Airways      2913
## 6 Virgin America   504
```

Summary: It appears that airlines get more negative tweets than positive and I struggle to understand why. I have flown quite a bit for work and for pleasure and it is rare that I have a negative experience. I truly enjoy flying and I often feel for flight attendants as they try their best to accommodate 100+ passengers on most typical flights. I wish persons would be a little more appreciative of the convenience of flying as opposed to having to drive or sail to destinations.

I really enjoyed sentimental analysis despite the issues that I had with my file not importing from github the way it does from my local drive and that I not successful referencing or citing the book in the first part of this exercise.