A Tale of Two Cities by Charles Dickens Wordcloud

Jorge Fitzmaurice

November 6, 2017

Abstract

In this article we construct a wordcloud, using the tidytext R package, for Charles Dicken's Novel A Tale of Two Cities.

A Tale of Two Cities (1859)[1] is a novel by Charles Dickens, set in London and Paris before and during the French Revolution. The novel tells the story of the French Doctor Manette, his 18-year-long imprisonment in the Bastille in Paris and his release to life in London with his daughter Lucie, whom he had never met; Lucie's marriage and the collision between her beloved husband and the people who caused her father's imprisonment; and Monsieur and Madame Defarge, sellers of wine in a poor suburb of Paris. The story is set against the conditions that led up to the French Revolution and the Reign of Terror.

1 The Gutenbergr Package

There is a relatively new package for R, guntenbergr, that gives one access to a variety of public domain works from the Project Gutenberg collection(**?**). One first has to install this package and bring it in with library. You may then call the following function to find the title number and then call the next function to download the document and store the result. The result will be a data frame.

```
library(gutenbergr)

## Warning:  package 'gutenbergr' was built under R version 3.4.2

gutenberg_works(title=='A Tale of Two Cities')

## # A tibble: 1 x 8
##   gutenberg_id                title           author gutenberg_author_id
##          <int>                <chr>            <chr>               <int>
## 1           98 A Tale of Two Cities Dickens, Charles                  37
## # ... with 4 more variables: language <chr>, gutenberg_bookshelf <chr>,
## #   rights <chr>, has_text <lgl>
```

[1] The novel was published by Charles Dickens.

```
dickens<-gutenberg_download(98)

dickens
```

```
## # A tibble: 15,865 x 2
##    gutenberg_id                              text
##           <int>                             <chr>
##  1           98             A TALE OF TWO CITIES
##  2           98
##  3           98 A STORY OF THE FRENCH REVOLUTION
##  4           98
##  5           98               By Charles Dickens
##  6           98
##  7           98
##  8           98                         CONTENTS
##  9           98
## 10           98
## # ... with 15,855 more rows
```

This dataframe has two columns, one for each line in Charles Dicken's Novel, and one indicating the gutenberg ID from which the book came from. Let's first unnest the lines to get every word in the novel in a different row:

```
library(tidytext)
library(dplyr)
dickens_words<-dickens%>%
                  unnest_tokens(word,text)
dickens_words$gutenberg_id<-NULL
head(dickens_words)
```

```
## # A tibble: 6 x 1
##     word
##    <chr>
## 1      a
## 2   tale
## 3     of
## 4    two
## 5 cities
## 6      a
```

Now we are ready to get the sentiments data frame.

2 The Sentiments Data Frame

The sentiments data frame is part of the tidy text package and it gives us different lexicons to work with. For this article the NRC lexicon will be used.

The NRC lexicon give us the emotion represented by certain word. For this article, we will only be using the fear words in the NRC lexicon for the first wordcloud and the joy words in the NRC lexicon for the second wordcloud.

```
nrc<-get_sentiments('nrc')
unique(nrc$sentiment)

##  [1] "trust"        "fear"         "negative"     "sadness"
##  [5] "anger"        "surprise"     "positive"     "disgust"
##  [9] "joy"          "anticipation"

fear_words<-nrc%>%
            filter(sentiment == 'fear')
head(fear_words)

## # A tibble: 6 x 2
##          word sentiment
##         <chr>     <chr>
## 1     abandon      fear
## 2   abandoned      fear
## 3 abandonment      fear
## 4   abduction      fear
## 5       abhor      fear
## 6   abhorrent      fear

joy_words<-nrc%>%
  filter(sentiment == 'joy')
head(joy_words)

## # A tibble: 6 x 2
##            word sentiment
##           <chr>     <chr>
## 1    absolution       joy
## 2     abundance       joy
## 3      abundant       joy
## 4      accolade       joy
## 5 accompaniment       joy
## 6    accomplish       joy
```

Next, we would like to join the data frame with the words for the novel and the data frame with the sentiments words to keep only those words that are represented in the sentiment lexicon with the selected sentiments.

3 The Wordcloud

To make the wordcloud, we first have to join the two data frames to get a new data frame with only the words with the appropriate sentiment are in there. We can use a function from the dplyr package for this:

```
dickens_fear<-inner_join(dickens_words,fear_words)

## Joining, by = "word"

dickens_fear

## # A tibble: 3,009 x 2
##          word sentiment
##         <chr>     <chr>
##  1 revolution      fear
##  2     honest      fear
##  3       plea      fear
##  4       fire      fear
##  5   darkness      fear
##  6        die      fear
##  7   darkness      fear
##  8    despair      fear
##  9       evil      fear
## 10      ghost      fear
## # ... with 2,999 more rows

dickens_joy<-inner_join(dickens_words,joy_words)

## Joining, by = "word"

dickens_joy

## # A tibble: 2,638 x 2
##              word sentiment
##             <chr>     <chr>
##  1 congratulatory       joy
##  2      companion       joy
##  3         honest       joy
##  4        triumph       joy
##  5           hope       joy
##  6        present       joy
##  7           good       joy
##  8        blessed       joy
##  9       birthday       joy
## 10        spirits       joy
## # ... with 2,628 more rows
```

Now, we need to calculate the frequencies of the words in the novel. Again, we can use standard dplyr techniques for this:

```
fear_freq<-dickens_fear%>%
                  group_by(word)%>%
                  summarize(count=n())
fear_freq
```

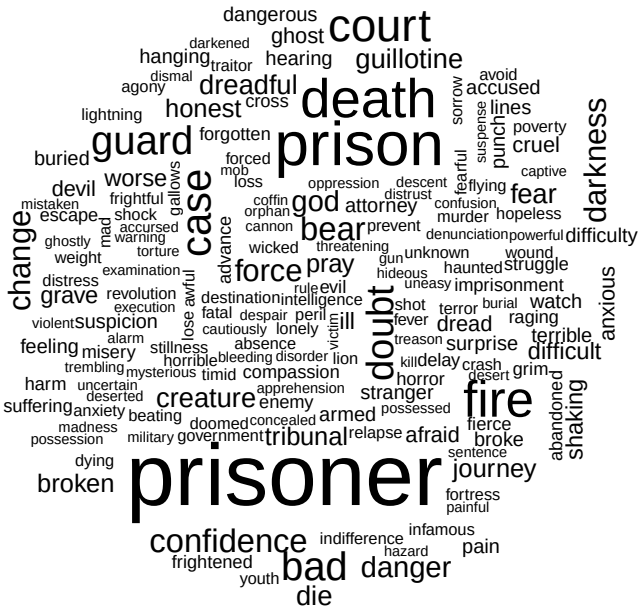
```
## # A tibble: 583 x 2
##           word count
##          <chr> <int>
##  1     abandon     1
##  2   abandoned    10
##  3 abandonment     1
##  4  abominable     2
##  5     absence     8
##  6       abuse     1
##  7       abyss     1
##  8    accident     3
##  9  accidental     3
## 10    accursed     5
## # ... with 573 more rows
```

```
joy_freq<-dickens_joy%>%
                  group_by(word)%>%
                  summarize(count=n())
joy_freq
```

```
## # A tibble: 335 x 2
##             word count
##            <chr> <int>
##  1     abundance     1
##  2      abundant     4
##  3 accompaniment     2
##  4  accomplished     5
##  5       achieve     1
##  6   achievement     1
##  7     admirable     4
##  8    admiration     5
##  9      adorable     1
## 10       advance    10
## # ... with 325 more rows
```

Finally, it's time to generate the wordcloud for the fear words:

```
library(wordcloud)
wordcloud(fear_freq$word,fear_freq$count,min.freq=5)
```



Lastly, it's time to generate the wordcloud for the joy words:

```
wordcloud(joy_freq$word,joy_freq$count,min.freq=3)
```

References

Fellows, I. (2014). *wordcloud: Word Clouds*. R package version 2.5.

Robinson, D. (2017). *gutenbergr: Download and Process Public Domain Works from Project Gutenberg*. R package version 0.1.3.

Robinson, D. and Silge, J. (2017). *tidytext: Text Mining using 'dplyr', 'ggplot2', and Other Tidy Tools*. R package version 0.1.4.

Wickham, H., Francois, R., Henry, L., and Mller, K. (2017). *dplyr: A Grammar of Data Manipulation*. R package version 0.7.2.

Xie, Y. (2017). *knitr: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.17.