Text Mining with R

Aojie Ju

11/18/2021

```
#Create a character vector.
text<-c("Because I could not stop for Death-",
"He kindly stopped for me-",
"The Carriage held but just Ourselves-",
"and Immortality")

text
```

```
## [1] "Because I could not stop for Death-"
## [2] "He kindly stopped for me-"
## [3] "The Carriage held but just Ourselves-"
## [4] "and Immortality"
```

```
#Turn it into a tidy text dataset.
library(dplyr)
```

```
##
## 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
```

```
text_df<-tibble(line=1:4,text=text) #tibble builds a data frame here.

text_df
```

```
## # A tibble: 4 x 2
##    line text
##   <int> <chr>
## 1     1 Because I could not stop for Death-
## 2     2 He kindly stopped for me-
## 3     3 The Carriage held but just Ourselves-
## 4     4 and Immortality
```

Keep in mind that a tibble is not compatible with tidy text analysis, since each row is made up of multiple combined words. So we need to convert this as a "one-token-per-document-per-row."

```
#break the text into individual tokens (tokenization) and transform it to a tidy data structure.
library(tidytext)
```

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

```
text_df %>%
  unnest_tokens(word,text)
```

```
## # A tibble: 20 x 2
##     line word
##    <int> <chr>
##  1     1 because
##  2     1 i
##  3     1 could
##  4     1 not
##  5     1 stop
##  6     1 for
##  7     1 death
##  8     2 he
##  9     2 kindly
## 10     2 stopped
## 11     2 for
## 12     2 me
## 13     3 the
## 14     3 carriage
## 15     3 held
## 16     3 but
## 17     3 just
## 18     3 ourselves
## 19     4 and
## 20     4 immortality
```

```
#Notice that unnest_tokens leaves out other columns, punctuations, and converts the tokens to lowercase
```

Then let's move on and do some additional tidying work. The "janeaustenr" package contains six novels of Jane Austen. The texts in a one-row-per-line format. We'll use "mutate()" to create columns linenumber and chapter.

```
#Construct the dataframe in one-row-per line format.
library(janeaustenr)
```

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

```
library(dplyr)
library(stringr)

original_books<-austen_books()%>%
  group_by(book)%>%
```

```
  mutate(linenumber=row_number(),
         chapter=cumsum(str_detect(text,
                                   regex("^chapter [\\divxlc]",
                                         ignore_case=TRUE)))) %>%
  ungroup()

original_books
```

```
## # A tibble: 73,422 x 4
##    text                    book                linenumber chapter
##    <chr>                   <fct>                    <int>   <int>
##  1 "SENSE AND SENSIBILITY" Sense & Sensibility          1       0
##  2 ""                      Sense & Sensibility          2       0
##  3 "by Jane Austen"        Sense & Sensibility          3       0
##  4 ""                      Sense & Sensibility          4       0
##  5 "(1811)"                Sense & Sensibility          5       0
##  6 ""                      Sense & Sensibility          6       0
##  7 ""                      Sense & Sensibility          7       0
##  8 ""                      Sense & Sensibility          8       0
##  9 ""                      Sense & Sensibility          9       0
## 10 "CHAPTER 1"             Sense & Sensibility         10       1
## # ... with 73,412 more rows
```

```
#Restructure it in one-token-per-row format.
library(tidytext)
tidy_books<-original_books%>%
  unnest_tokens(word,text)

tidy_books
```

```
## # 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
```

In many cases, we remove stop words. In package "tidytext," we have a dataset "stop_words" with an "anti_join()."

```
data(stop_words)

tidy_books<-tidy_books%>%
  anti_join(stop_words)
```

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

How can we count the common words in all of the book? We use “count()” in package “dplyr.”

```
tidy_books%>%
  count(word, sort=TRUE)
```

```
## # A tibble: 13,914 x 2
##    word       n
##    <chr>  <int>
##  1 miss    1855
##  2 time    1337
##  3 fanny    862
##  4 dear     822
##  5 lady     817
##  6 sir      806
##  7 day      797
##  8 emma     787
##  9 sister   727
## 10 house    699
## # ... with 13,904 more rows
```