Using tidytext to Clean and Analyze Text

Author: Jason Chen

Theme: Data Manipulation

Introduction

So far in this class, we have done a lot of data analysis on numeric data. Through tidyverse packages such as dplyr and ggplot we have been able to manipulate this data with greater ease and convenience. However, we have done less with textual data. While we have learned how to manipulate individual strings, doing this on large textual datasets is clumsy and impractical. The tidytext package is a tidyverse package in the vein of dplyr that makes handling textual data easier. It also contains datasets of specific types of words that makes cleaning and analyzing data much easier. This allows to easily do more involved analysis on text, as we can already do on numeric data. One of the more interesting analyses we can do is sentiment analysis. tidytext contains tables of words with either positive or negative connotations. We can count the number of occurences of these words in the text to calculate a general sentiment of the text.

tidyverse Packages

The “tidyverse” is a collection of R packages built for data analysis, designed with a shared philosophy, that data is organized into similar tables with few columns, which makes working with the data easier. Because tidy packages share similar data representations, they are easily used in tandem. For example, dplyr makes working with data frames easier, and these data frames can then be easily graphed with ggplot2. tidytext is another tidy package that allows us to work with text more easily and more consistently with other tidy packages.

External Packages

We obviously must import the tidytext package. We also need readtext to read int .txt files, and stringr to perform basic string manipulations, and dplyr and ggplot2 to manipulate tidy dataframes and graph them.

```
library(stringr)
library(tidytext)
```

```
## Warning: package 'tidytext' was built under R version 3.4.2
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.4.2
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

```
##
## 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
```

```
library(ggplot2)
library(readtext)
```

tidytext Tutorial

1. First, we need to find an appropriate dataset to analyze. On Project Gutenburg, we can download .txt documents of famous novels. Here, I have downloaded The Metamorphosis by Franz Kafka. The link to the texts used in this tutorial can be found in the References section at the bottom.

```
t = readLines("pg5200.txt")

kafka <- data.frame(linenumber = 1:2362)
kafka$text = t
```

2. The unest_tokens method in tidytext allows us to take a table of text organized by line and break into a table of individual words.

```
tidy_kafka <- kafka %>%
  unnest_tokens(word, text)

head(tidy_kafka)
```

```
##     linenumber          word
## 1            1           the
## 1.1          1       project
## 1.2          1     gutenberg
## 1.3          1         ebook
## 1.4          1            of
## 1.5          1 metamorphosis
```

3. The package contains list of unessential words, which can be called with “stop_words”. We can use anti_join to create a new table excluding these words.

```
data("stop_words")
cleaned_kafka <- tidy_kafka %>%
  anti_join(stop_words)
```

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

4. From this new table we can count the most common essential words in the text.

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

```
## # A tibble: 2,572 x 2
##         word     n
##        <chr> <int>
##  1    gregor   199
##  2  gregor's    99
##  3    father    96
##  4    sister    96
##  5 gutenberg    88
##  6   project    88
##  7      door    87
##  8    mother    82
##  9      time    59
## 10        tm    56
## # ... with 2,562 more rows
```

5. nrc is a table of words with associated sentiment values. Sentiment is the positivity or negativity of certain words, and by counting these we can see the general mood of the text. We can use semi_join to count the most common “joy”-associated words that occur in the text.

```
nrc <- get_sentiments("nrc") %>%
  filter(sentiment == "joy")

tidy_kafka %>%
  semi_join(nrc) %>%
  count(word, sort = TRUE)
```

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

```
## # A tibble: 102 x 2
##       word     n
##      <chr> <int>
##  1  mother    82
##  2    good    19
##  3   found    16
##  4   money    16
##  5    food    14
##  6 finally    10
##  7 present     9
##  8    kind     8
##  9   peace     8
## 10   music     7
## # ... with 92 more rows
```

6. bing is another table of sentiment value. We can use bing to detemine the overall sentiment of each portion of the text.

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

kafkasentiment <- tidy_kafka %>%
  inner_join(bing) %>%
  count(index = linenumber %/% 80, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative)
```

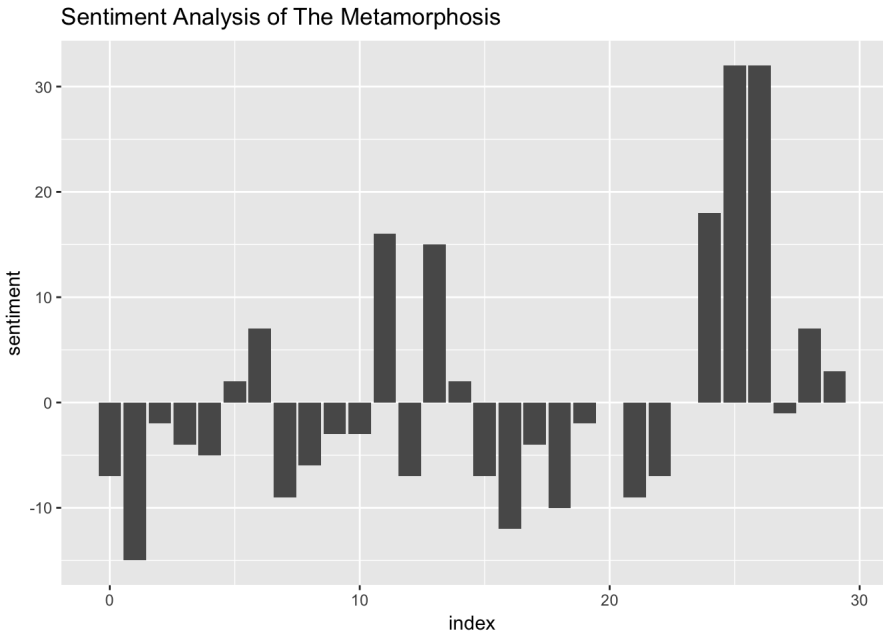
```
## Joining, by = "word"
```

```
kafkasentiment
```

```
## # A tibble: 30 x 4
##    index negative positive sentiment
##    <dbl>    <dbl>    <dbl>     <dbl>
##  1     0       15        8        -7
##  2     1       36       21       -15
##  3     2       31       29        -2
##  4     3       31       27        -4
##  5     4       33       28        -5
##  6     5       19       21         2
##  7     6       25       32         7
##  8     7       25       16        -9
##  9     8       32       26        -6
## 10     9       21       18        -3
## # ... with 20 more rows
```

7. We can use to these sentiment values to break each part of the text into its own positivity value. We can then use ggplot to graph a bar chart of positivity at each point in the text.

```
ggplot(kafkasentiment, aes(x = index, y = sentiment)) + geom_bar(stat = "identity", show.legend = FALSE) + ggtitle
("Sentiment Analysis of The Metamorphosis")
```



8. We can also sum up the sentiment rows to find a total sentiment value of the text.

```
sum(kafkasentiment$sentiment)
```

```
## [1] 21
```

9. We can do the same procedure on another text (The Adventures of Tom Sawyer by Mark Twain). We can see the chart of sentiment for this novel as well.

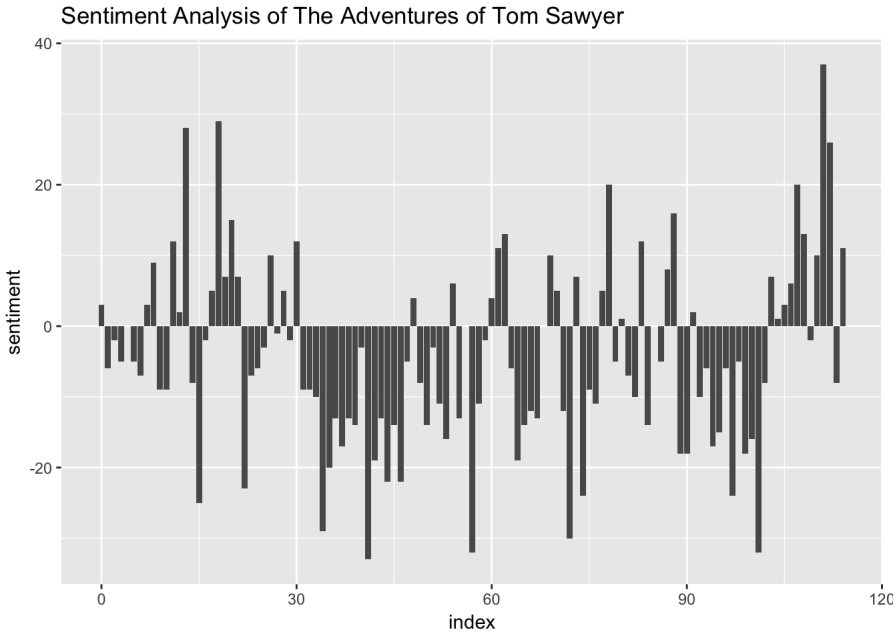
```
t2 = readLines("74-0.txt")

tomsawyer <- data.frame(linenumber = 1:9209)
tomsawyer$text = t2

tidy_tomsawyer <- tomsawyer %>%
  unnest_tokens(word, text)
cleaned_tomsawyer <- tidy_tomsawyer %>%
  anti_join(stop_words)
```

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

```
tomsawyersentiment <- tidy_tomsawyer %>%
  inner_join(bing) %>%
  count(index = linenumber %/% 80, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative)
```

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

```
ggplot(tomsawyersentiment, aes(x = index, y = sentiment)) + geom_bar(stat = "identity", show.legend = FALSE) + ggt
itle("Sentiment Analysis of The Adventures of Tom Sawyer")
```



10. Once again, we can sum up the sentiment values of the entire novel. Surprisingly, it seems that Tom Sawyer is overall much more negative than The Metamorphosis.

```
sum(tomsawyersentiment$sentiment)
```

```
## [1] -484
```

Conclusion

In conclusion, we can see that the tidytext package makes analyzing text much easier. It is used to easily break texts into individual word tokens, and has handy tables of sentimental words. These tables make sentiment analysis easy with simple join and antijoins. The tidytext package also follows tidy principles and therefore is easily used alongside other tidyverse packages. *tidytext is yet another tool an R user has in his toolbox, this one to make handling textual data easier and basic sentiment analysis possible.*

References

Project Gutenberg - The Adventures of Tom Sawyer - https://www.gutenberg.org/ebooks/74

Project Gutenberg - The Metamorphosis - http://www.gutenberg.org/ebooks/5200

CRAN - Introduction to tidytext - https://cran.r-project.org/web/packages/tidytext/vignettes/tidytext.html

RStudio - What is tidyverse - https://rviews.rstudio.com/2017/06/08/what-is-the-tidyverse/

RStudio - Managing Unstructured Data with the tidytext package - https://rstudio-pubs-static.s3.amazonaws.com/235472_460013007fa74af191c58ba4305ba170.html

Rbloggers - Sentiment Analysis on Donald Trump using R - https://www.r-bloggers.com/sentiment-analysis-on-donald-trump-using-r-and-tableau/

Tidytextmining - Sentiment analysis with tidy data - http://tidytextmining.com/sentiment.html

I'm Jacob - Using tidytext to make sentiment analysis easy - http://jacobsimmering.com/2016/11/15/tidytext/