Using R to Create a Wordcloud from e-book Text

Judy Minichelli

November 7, 2017

Abstract

This article gives instructions on how to download text from an e-book and create a wordcloud in R. A wordcloud is a data visualization that shows the most commonly used words in a large text dataset where word size is proportional to frequency; words with the highest count appear larger and in bold.

Wordclouds have been used to summarize survey results and show most popular dog breeds in the US. Certainly other text datasets can be quickly distilled down to a simple graphic (see example at end of article) to effectively communicate with an audience.

In this example, the text data set used to create a wordcloud was extracted from the e-book *The Wonderful Wizard of Oz*[1]. Written by Frank Baum and published in 1900, it was the basis for the 1902 Broadway musical and the 1939 classic movie starring Judy Garland.

1 Required Packages for R

The following packages were installed and brought in with library; dplyr, gutenbergr, stringr, tidytext, tm, and wordcloud.

The gutenbergr package is used to download e-book text. *The Wonderful Wizard of Oz* is e-book number 55 as identified in the book's bibrec tab on the Project Gutenberg website. R can also be used to locate an e-book number by using: gutenberg_works(title=='enter title here'). Or, if the exact title in not known, use a key word search: gutenberg_works(title==str_detect(title,'enter key word here')).

The code below stores the result of the book text download in the data frame "wizard".

```
library(gutenbergr)
wizard<-gutenberg_download(55)
```

The "wizard" data frame contains two columns; the line reference number and the corresponding text from the book.

[1] https://en.wikipedia.org/wiki/The$_{W}onderful_{W}izard_{o}f_{O}z$.

2 How to Clean the Data

For this exercise, it is not necessary to exclude chapter headings and the first few pages of text that are not part of the story; however, the procedure below accomplishes the task. First, the stringr string detect command filters out chapter headings.

```
library(stringr)
library(dplyr)
wizard<-wizard%>%
  filter(!str_detect(wizard$text,'^CHAPTER'))
```

Then the wizard data frame is redefined to exclude the pre-story text lines.

```
wizard<-wizard[36:4721,]
```

3 Creating the Wordcloud

To make the wordcloud, first read the wizard data frame into the words_df data frame and then use the unnest_tokens comand to break the text lines into individual words.

```
library(tidytext)
words_df<-wizard%>%
  unnest_tokens(word,text)

words_df

## # A tibble: 39,557 x 2
##    gutenberg_id         word
##           <int>        <chr>
##  1           55           24
##  2           55         home
##  3           55        again
##  4           55 introduction
##  5           55     folklore
##  6           55      legends
##  7           55        myths
##  8           55          and
##  9           55        fairy
## 10           55        tales
## # ... with 39,547 more rows
```

Filter out commonly used ”generic” words from the data frame like articles, prepositions, and pronouns, for example ”the”, ”after”, and ”you” using the stop_words command.

```
words_df<-words_df%>%
  filter(!(word %in% stop_words$word))

words_df

## # A tibble: 12,223 x 2
##    gutenberg_id         word
##           <int>        <chr>
##  1           55           24
##  2           55         home
##  3           55 introduction
##  4           55     folklore
##  5           55      legends
##  6           55        myths
##  7           55        fairy
##  8           55        tales
##  9           55    childhood
## 10           55         ages
## # ... with 12,213 more rows
```

Calculate the frequencies of the remaining unique words.

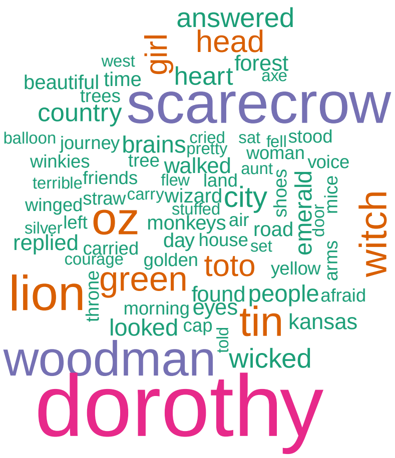
```
word_freq<-words_df%>%
  group_by(word)%>%
  summarize(count=n())

word_freq

## # A tibble: 2,507 x 2
##     word count
##    <chr> <int>
##  1     1     1
##  2    10     1
##  3    11     1
##  4    12     1
##  5    13     1
##  6    14     1
##  7    15     1
##  8    16     1
##  9    17     1
## 10    18     1
## # ... with 2,497 more rows
```

Generate the wordcloud. If there are too many or too few words, adjust ”n” in ”min.freq=n” to change the minimum number of occurrences required for the word to appear in the wordcloud. The number of colors used in the graphic can be changed by adjusting "n" in "colors=brewer.pal(n,'Dark2').

Additionally, a different color scheme may be selected by changing "Dark 2" to another R Color Palette; a Google search will yield a variety of references and "cheat sheets" to help select an alternate[2].

Although black and white is striking, using a few different colors helps further differentiate the words used with higher frequency and makes the graphic more appealing.

```r
library(wordcloud)
library(tm)
wordcloud(word_freq$word,word_freq$count,min.freq=25,colors=brewer.pal(4,'Dark2'))
```



[2] https://www.nceas.ucsb.edu/ frazier/RSpatialGuides/colorPaletteCheatsheet.pdf

4 Results and Conclusion

Each time the wordcloud is generated, the words change position and orientation but word size remains constant as it is dependent on frequency which does not change.

Being the main character, of course Dorothy has top-billing. And it follows that characters Scarecrow and (Tin) Woodman would be tied for second, Oz and Lion third, and then Witch and Toto fourth. The ever fabulous munchkins barely made mention and are still protesting.

References

Baum, F. (1900). *The Wonderful Wizard of Oz.* George M. Hill Company.

Feinerer, I. and Hornik, K. (2017). *tm: Text Mining Package.* R package version 0.7-1.

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

Robinson, D. (2017a). *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.

Robinson, J. S. . D. (2017b). *Text Mining with R.* O'Reilly.

Wickham, H. (2017). *stringr: Simple, Consistent Wrappers for Common String Operations.* R package version 1.2.0.

Wickham, H., Francois, R., Henry, L., and Mller, K. (2017). *dplyr: A Grammar of Data Manipulation.* R package version 0.7.4.