Wordcloud Analysis of Dr. Jekyll and Mr. Hyde

Ron Richardson

October 30, 2017

Abstract

This article contains the code and explanation of how to create a wordcloud using the text from Robert Louis Stevenson's novel *The Strange Case of Dr. Jekyll and Mr. Hyde*.

1 Introduction

A wordcloud is a type of visualization that shows the frequency of a word within a collection by changing the size of the word. The more frequently a word is used, the larger it shows up in the wordcloud. In this article, we will be using R and RStudio to create a wordcloud.

In the spirit of Halloween, we have chosen the novel *The Strange Case of Dr. Jekyll and Mr. Hyde* by Robert Louis Stevenson. It was written in 1886 and is available via Project Gutenberg, who graciously provide the text via their R package gutenbergr.

```
What is Captain Blackbeard`s favorite programming language?

ARRRRR
```

2 Obtaining the Text

To obtain the text, we must first import our libraries and find the text we want. For the purposes of this article, we have already searched for the text we want, but have included code to search by title.

```
library(gutenbergr)
library(dplyr)
library(stringr)

#search by title
gutenberg_works(str_detect(title, 'Jekyll'))
```

```
## # A tibble: 1 x 8
##   gutenberg_id                                        title
##          <int>                                        <chr>
## 1           42 The Strange Case of Dr. Jekyll and Mr. Hyde
## # ... with 6 more variables: author <chr>, gutenberg_author_id <int>,
## #   language <chr>, gutenberg_bookshelf <chr>, rights <chr>,
## #   has_text <lgl>

#store the text into a data frame of two columns
jh<-gutenberg_download(42, strip=TRUE)
```

3 Cleaning the Text

Once we have downloaded our text, we want to remove any lines of text that contain information we don‘t want to include in our wordcloud. This can contain information such as the label for each chapter, or the title page and author‘s name.

By looking at the first 20 rows, we can see the first 10 contain the title page. We can also se on row 11, how the chapter labels are formatted. Using a regular expression, we can filter those rows out and remove them from our data frame.

```
#remove title page
jh<-jh[10:dim(jh),]

#remove chapter labels
jh<-jh%>%
  filter(str_detect(jh$text, "^\\d+\\)")==FALSE)
```

4 Creating Words From Lines

To create a wordcloud, we need to know the frequency of each word. Currently, our data is stored in lines, so we need to convert them. The tidytext package provides a great function to un-nest the words. This will break apart each line (the ‘text‘ column) into words, which we will store into a new dataframe.

```
library(tidytext)

words<-jh%>%
  unnest_tokens(word, text)
```

Once we have the words split apart, we want to remove any unnecessary words, such as indefinite articles (a, an, the). Again, the tidytext package provides a function that contains these words, called stop words.

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

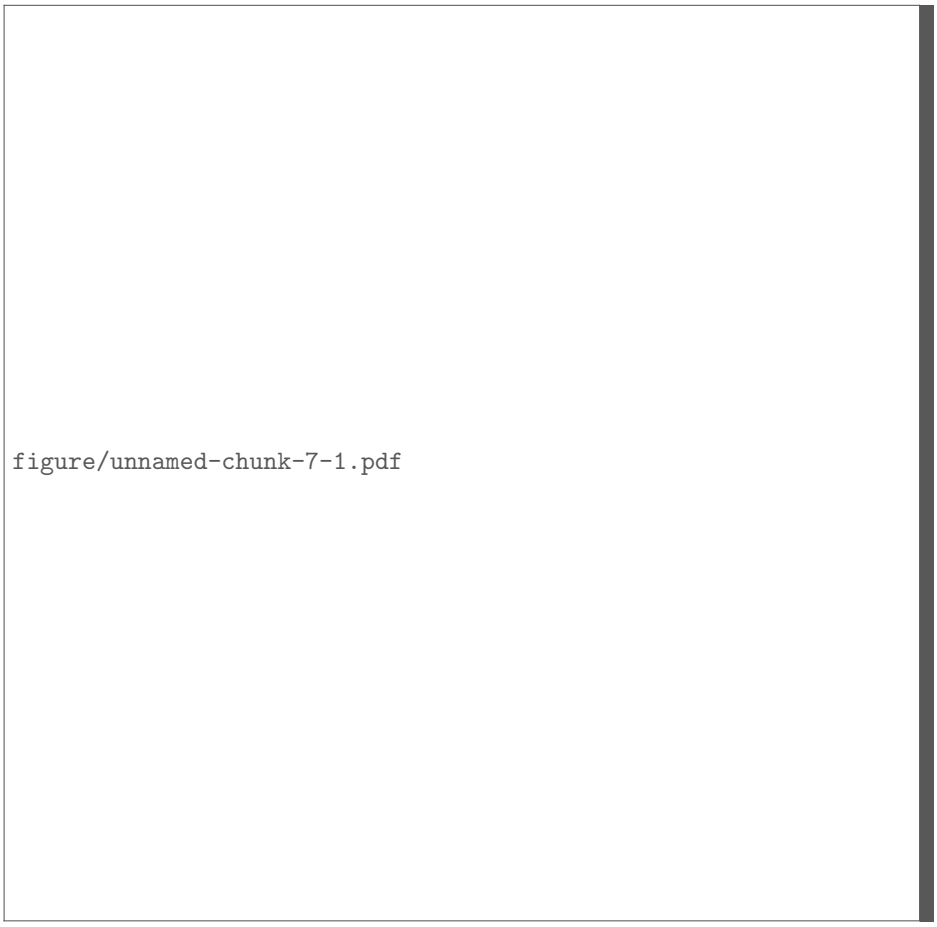
5 Creating the Wordcloud

Now that we have a dataframe of just words, we need to get a count of how many times each word is used. Using the package dplyr, we can run a quick grouping query into a new dataframe.

```
freq<-words%>%
  group_by(word)%>%
  summarize(count=n())
```

The wordcloud package contains many customizations, but for the purposes of this article, we will keep it simple. We want to show the top 100 most used words in our wordcloud. There are over 3000 unique words in the novel, so generating the wordcloud could take a lengthy time if we included each one.

```
library(wordcloud)
wordcloud(freq$word, freq$count, max.words=100)
```



6 Summary

Looking at the wordcloud above, we can easily see which words are frequently used. In the case of this novel, it appears that the words ‘utterson‘, ‘jekyll‘, and ‘hyde‘ are the most frequent, with ‘poole‘, ‘lawyer‘, and ‘sir‘ coming in close behind.

Just from the wordcloud, we can surmise that ‘utterson‘ is a person‘s name and ‘jekyll‘ and ‘hyde‘ are also character‘s in the novel. Another hypothesis is that one of them is a laywer.

We can see now that generating a wordcloud is quite easy and can be a useful visualization to get a quick sense of what a collection of text is about.