STAT 209
Text Data

August 17, 2021

Colin Reimer Dawson

Working With Text

- The fields of *computational linguistics* and *natural language processing* (NLP) develop methods for extracting patterns and making predictions using **text data**
- Example applications
  - Automated question answering
  - Information retrieval
  - Speech recognition (involves both text analysis and signal processing)
- Our focus
  - Before *modeling* text, need to do some *preprocessing* and compute *statistics*

Outline

Shakespeare

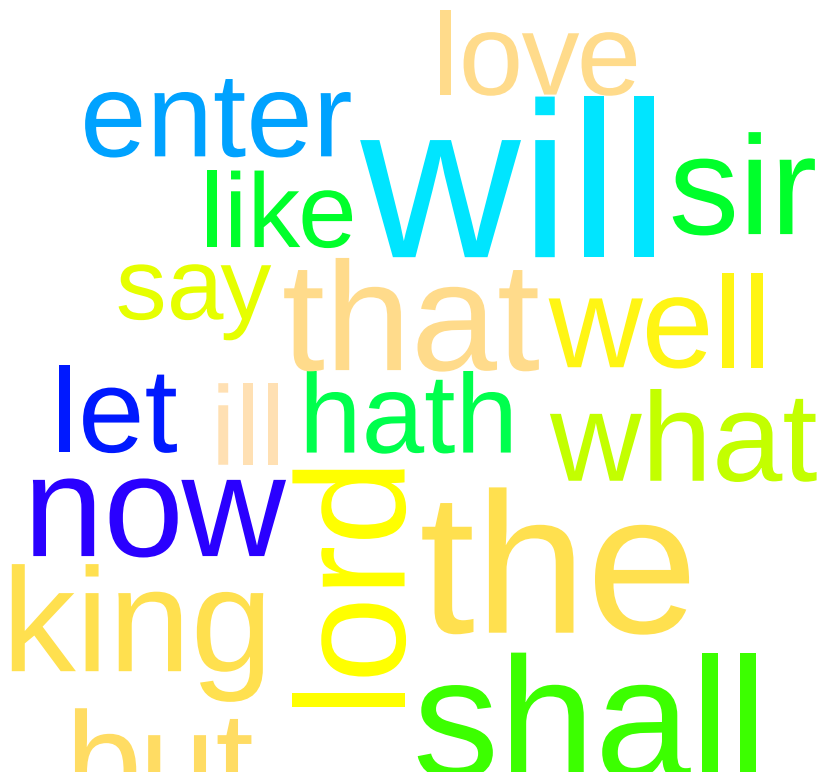
```
library(tidyverse); library(RCurl); library(magrittr)
url1 <- "https://ocw.mit.edu/ans7870/6/6.006/s08/lecturenotes/files/"
url2 <- "t8.shakespeare.txt"
url <- paste0(url1, url2)
shakespeare <- getURL(url)
## `shakespeare` contains a single ginormous string
## Can split it into fixed length substrings to view a piece of it
shakespeare %>% strwrap() %>% extract(1000:1010)
```

```
 [1] "I in your sweet thoughts would be forgot, If thinking on me then"
 [2] "should make you woe.  O if (I say) you look upon this verse, When"
 [3] "I (perhaps) compounded am with clay, Do not so much as my poor"
 [4] "name rehearse; But let your love even with my life decay.  Lest"
 [5] "the wise world should look into your moan, And mock you with me"
 [6] "after I am gone."
 [7] ""
 [8] "72 O lest the world should task you to recite, What merit lived in"
 [9] "me that you should love After my death (dear love) forget me"
[10] "quite, For you in me can nothing worthy prove.  Unless you would"
[11] "devise some virtuous lie, To do more for me than mine own desert,"
```

Finding the most common words

An easy and fun (albeit statistically questionable) visualization of text is the **word cloud**

```
library(wordcloud); library(tm)
shakespeare %>% wordcloud(
    max.words = 30, scale = c(8, 1),
    colors = topo.colors(n = 30), random.color = TRUE)
```



Preprocessing

- Most common words are grammatical words (NLP people call them "stop words": a, the, of, etc.). Not that interesting.
- Also may be more than one version of each word (capitalized, etc.)
- Punctuation and extra whitespace can muddy things up as well

Preprocessing

```
library(tm) # text-mining package
shakespeare_corpus <- shakespeare %>%
    VectorSource() %>%           # change the data type
    VCorpus() %>%                # change the data type
    tm_map(stripWhitespace) %>% # remove spaces/line-breaks, etc.
    tm_map(removeNumbers) %>%   # remove line numbers, etc.
    tm_map(removePunctuation) %>% # what it says
    tm_map(content_transformer(tolower)) %>% # all lowercase
    tm_map(removeWords, stopwords("english")) # remove function words
```

Word Cloud Again

```
shakespeare_corpus %>% wordcloud(
    max.words = 30, scale = c(8, 1),
    colors = topo.colors(n = 30), random.color = TRUE)
```