Creating Word Clouds of Thesis & Dissertation Titles

Clarke Iakovakis

April 2017

Introduction

The purpose of this document is to describe how word cloud visualizations were created for the titles of master's projects, theses, and dissertations at the University of Houston-Clear Lake.

Text Mining

```
tds <- read.csv("./data/results/TDCollectionWithFaculty-or.csv"
                , sep = ","
                , stringsAsFactors = F
                #, quote = "`"
                , na.strings = c("NA", "")
                , header = TRUE)
```

The data was pulled from the library catalog and returned to a dataframe `tds`.

Titles in the catalog include the author's name followed by a forward slash, e.g. . Therefore the first step is to extract all text before that slash. This is done using the `str_extract` and `str_detect` functions from the **stringr** package. After that, the whole list of titles is collapsed into one long, single vector and coerced to a character vector.

```
titles <- str_extract(tds$TITLE[which(str_detect(tds$TITLE, "\\/"))], "^.*?(?=\\/)")
titles <- paste(unlist(titles), collapse = " ")
titles <- as.character(titles)
```

Next, the string is converted to a corpus ready for text mining, using the `Corpus` function from the **tm**.

Much of the following code was adapted from the post, "Text mining and word cloud fundamentals in R : 5 simple steps you should know."

A short `toSpace` custom function is written converting all punctuation to a space.

```
titlesCorp <- Corpus(VectorSource(titles))
toSpace <- content_transformer(function (x , pattern ) str_replace_all(x, pattern, " "))
titlesCorp <- tm_map(titlesCorp, toSpace, "[[:punct:]]")
```

Then, it is converted to lowercase and numbers, stop words, punctuation, and whitespace are removed.

```
titlesCorp <- tm_map(titlesCorp, content_transformer(tolower))
titlesCorp <- tm_map(titlesCorp, removeNumbers)
titlesCorp <- tm_map(titlesCorp, removeWords, stopwords("english"))
titlesCorp <- tm_map(titlesCorp, removePunctuation)
titlesCorp <- tm_map(titlesCorp, stripWhitespace)
```

Finally, the string is split by space (each word is made a value). Since `strsplit` splits it into lists, the following command takes the entirety of the first list, from the second value on. It is then coerced to a dataframe.

Then, because so many single values were created (from middle initials, etc.), create a dataframe z of the number of characters in each line, and use the `which` command to subset them out.

```
t <- strsplit(as.character(titlesCorp), " ")  # split
t <- t[[1]][2:length(t[[1]])]

t <- data.frame("titlewords" = t, stringsAsFactors = F)

z <- data.frame("TITLENCHAR" = apply(t, 1, nchar))
TITLEWORDS = t$titlewords[-which(z$TITLENCHAR == 1)]
t <- data.frame(TITLEWORDS
                , "COLLEGE" = rep(tds$College[1], length(TITLEWORDS))
                , stringsAsFactors = F)
```

This code was run on subsets of each of the four colleges, to create a word corpus by college.

Visualizing the data in Tableau

Each of the four datasets was imported into Tableau. In essence, I followed the instructions at the post Text Data in Tableau to create the wordcloud visualization. The primary difference is that stop words were already removed.

View the visualization at: http://libguides.uhcl.edu/thesesdissertations/titlewordcloud