Milestone Report

Brian Perron

This report is submitted in partial fulfillement of the Data Science Specialization Swiftkey Capstone. The focus of the capstone project involves applying data science to the area of Natural Language Processing. Students are provided with text data from a corpus called HC Corpora (from www.corpora.heliohost.org). Using these data, students are expected to: 1) Develop a predictive text model that is tested on a real data set; 2) Create a reproducible R markdown document describing your model building process; and 3) Build a Shiny or Yhat application to demonstrate the use of the product.

This milestone report summarizes the major tasks and activities to date, including data acquisition and cleaning, and exploratory analysis of three English data files. The main body of the report is limited to approximately two pages. An Appendix containing additional code, output and graphics is available to provide more in-depth documentation of the work to date.

Data acquisition and cleaning

As a first step in acquiring the data, I wanted to know some important features of the data, including the file size and number of lines. While this can be done in R, it is much more effecient to answer these questions with the Mac Terminal (i.e., Unix commands). The following are example commands used to obtain information about these files. This assumes the commands are invoked while in the directory where the files are located.

```
# Count number of bytes, lines, and words (respectively) for each file
wc -c -l -w en_US.blogs.txt en_US.twitter.txt en_US.news.txt
    899288 37334626 210159816 en_US.blogs.txt
   2360148 30374206 167105338 en_US.twitter.txt
   1010242 34372720 205811889 en_US.news.txt
  4269678 102081552 583077043 total
```

The next step involved initializing the workspace and reading the data into R. As I was only interested in setting up procedures to clean the data, I reduced the overall file by specifying the *n*-size in the `readLines` function. It should be noted that I am simply building data cleaning procedures and not analyzing the data. Thus, randomly divided test and training data are not necessary at this point.

```
rm(list = ls())
libs <- c("tm", "SnowballC", "XML", "ggplot2", "wordcloud", "tau")
lapply(libs, require, character.only = TRUE)

# Set the following path to the location of the text file blogs <-
# readLines('~/Git/capstoneCoursera/en_US/en_US.blogs.txt') news <-
# readLines('~/Git/capstoneCoursera/en_US/en_US.news.txt')
tweets <- readLines("~/Git/capstoneCoursera/en_US/en_US.twitter.txt", n = 100)
```

To facilitate the cleaning of three different files (as opposed to merging them into one large file), I created a simple function as a wrapper for a series of functions. My rules for cleaning and pre-processing followed the general procedures in Natural Language Processing (NLP) research.

```
cleaner.f <- function(x, print=FALSE){
     x <- lapply(x, function(row) iconv(row, "latin1", "ASCII", sub=""))
     x <- tolower(x); x <- removeWords(x, stopwords("english"))
     x <- stemDocument(x); x <- sub("^\\s+", "", x)
     x <- removeNumbers(x); x <- removePunctuation(x)
     x <- sub("\\<rt\\>", "", x); x <- stripWhitespace(x)
     return(unlist(x))}
tweets <- cleaner.f(tweets);
```

Data Exploration and Visualization

After cleaning these data, I created a corpus to facilitate exploration of the text data and *term document* and *document term* matrices. From this I was able to examine the frequencies of different words, associations, and distributions.

freq[head(ord)]; freq[tail(ord)]

length(freq)

Least frequent terms

Most frequent terms