Peer-graded Assignment Milestone Report

Noha Mohamed

16 Oct, 2018

Basic summary

This report provides a short overview of the exploratory analysis of the text data to be used for the Capstone project for the Data Science Specialization along with a description of plans for the word prediction algorithm.

Tasks to accomplish

1. Demonstrate that you've downloaded the data and have successfully loaded it in.
2. Create a basic report of summary statistics about the data sets.
3. Report any interesting findings that you amassed so far.
4. Get feedback on your plans for creating a prediction algorithm and Shiny app

Data Analysis

After download the file from
Coursera: https://d396qusza40orc.cloudfront.net/dsscapstone/dataset/Coursera-SwiftKey.zip

1. Load the R packages necessary for running the analysis
2. Building a table

```
3. file.list = c("C:/Users/CEO/Documents/10 - Capstone Project/Peer-graded
   Assignment Milestone Report/Coursera-SwiftKey/final/en US/en US.blogs.t
   xt", "C:/Users/CEO/Documents/10 - Capstone Project/Peer-graded Assignme
   nt Milestone Report/Coursera-SwiftKey/final/en_US/en_US.news.txt", "C:/
   Users/CEO/Documents/10 - Capstone Project/Peer-graded Assignment Milest
   one Report/Coursera-SwiftKey/final/en_US/en_US.twitter.txt")
4.
5. text <- list(blogs = "", news = "", twitter = "")
6.
7. matrix.summary <- matrix(0, nrow = 3, ncol = 3, dimnames = list(c("blog
   s", "news", "twitter"),c("file size, Mb", "lines", "words")))
8. for (i in 1:3) {
9.   con <- file(file.list[i], "rb")
10.    text[[i]] <- readLines(con, encoding = "UTF-8",skipNul = TRUE)
```

```
    close(con)
    matrix.summary[i,1] <- round(file.info(file.list[i])$size / 1024^2,
  2)
    matrix.summary[i,2] <- length(text[[i]])
    matrix.summary[i,3] <- sum(stri_count_words(text[[i]]))
}
```

```
kable(matrix.summary)
```

	File size, Mb	*lines*	*words*
Blogs	200.42	899288	37546246
News	196.28	1010242	34762395
twitter	159.36	2360148	30093410

How the files are very large, we will proceed with the analysis using a small fraction to get a sample. For example, News file is 196MB of size and 1.010,242 Lines. I will use 5k random lines for analysis.

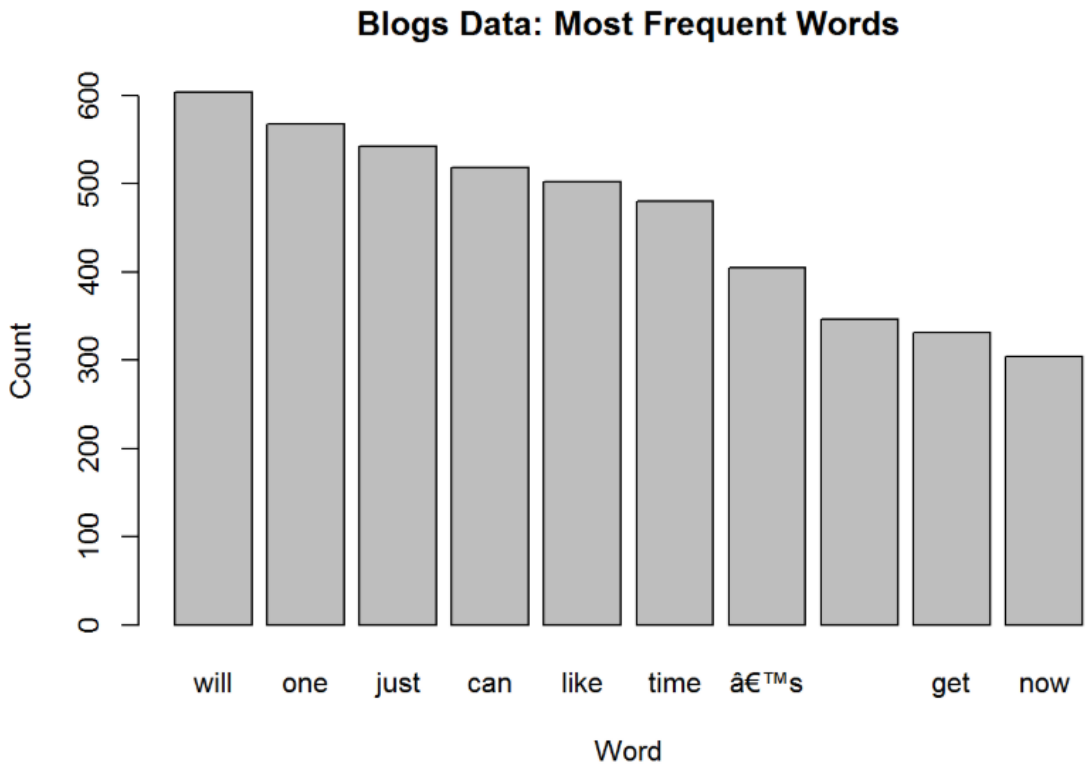
```
set.seed(123)
blogs_sample <- sample(text$blogs, 0.005*length(text$blogs))
news_sample <- sample(text$news, 0.005*length(text$news))
twitter_sample <- sample(text$twitter, 0.005*length(text$twitter))
```

Blogs Sample

```
# Create corpus
corpus1 <- Corpus(VectorSource(blogs_sample))
# To lower case
corpus1 <- tm_map(corpus1, content_transformer(tolower))
# Remove punctuation marks
corpus1 <- tm_map(corpus1, removePunctuation)
# Remove numbers
corpus1 <- tm_map(corpus1, removeNumbers)
#remove stop words
```

```
corpus1 <- tm_map(corpus1, removeWords, stopwords("english"))
#Remove whitespaces
corpus1 <- tm_map(corpus1, stripWhitespace)
```

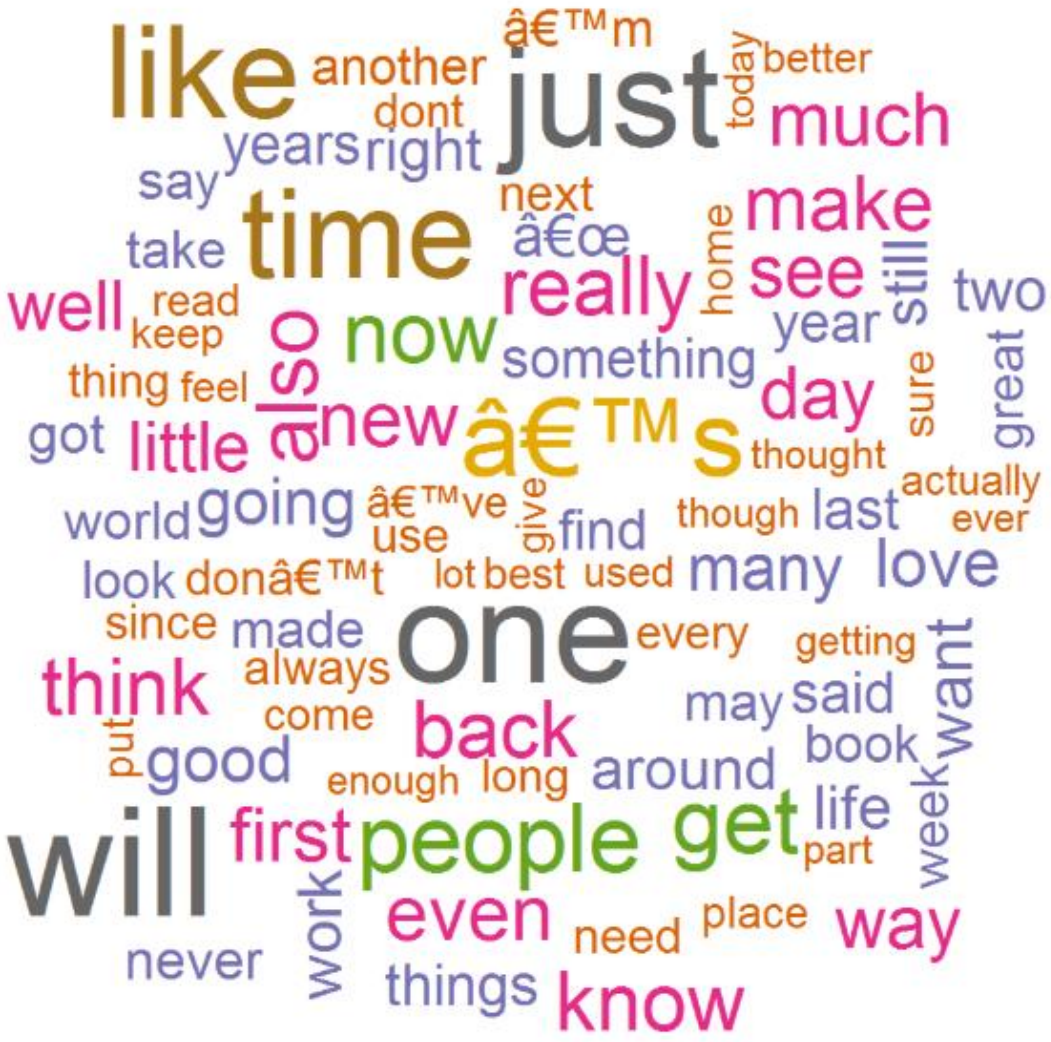
```
frequentWords <- head(sort(rowSums(as.matrix(TermDocumentMatrix(corpus1))),de
creasing=TRUE), 10)


barplot(frequentWords,
        main = "Blogs Data: Most Frequent Words",
        xlab="Word",
        ylab = "Count")
```




```
term.doc.matrix1 <- TermDocumentMatrix(corpus1)
term.doc.matrix1 <- as.matrix(term.doc.matrix1)
word.freqs1 <- sort(rowSums(term.doc.matrix1), decreasing=TRUE)
dm1 <- data.frame(word=names(word.freqs1), freq=word.freqs1)
```

```
wordcloud(dm1$word, dm1$freq, min.freq= 100, random.order=TRUE, rot.per=.25,
colors=brewer.pal(8, "Dark2"))
```

```
## Warning in wordcloud(dm1$word, dm1$freq, min.freq = 100, random.order =
## TRUE, : can could not be fit on page. It will not be plotted.
```

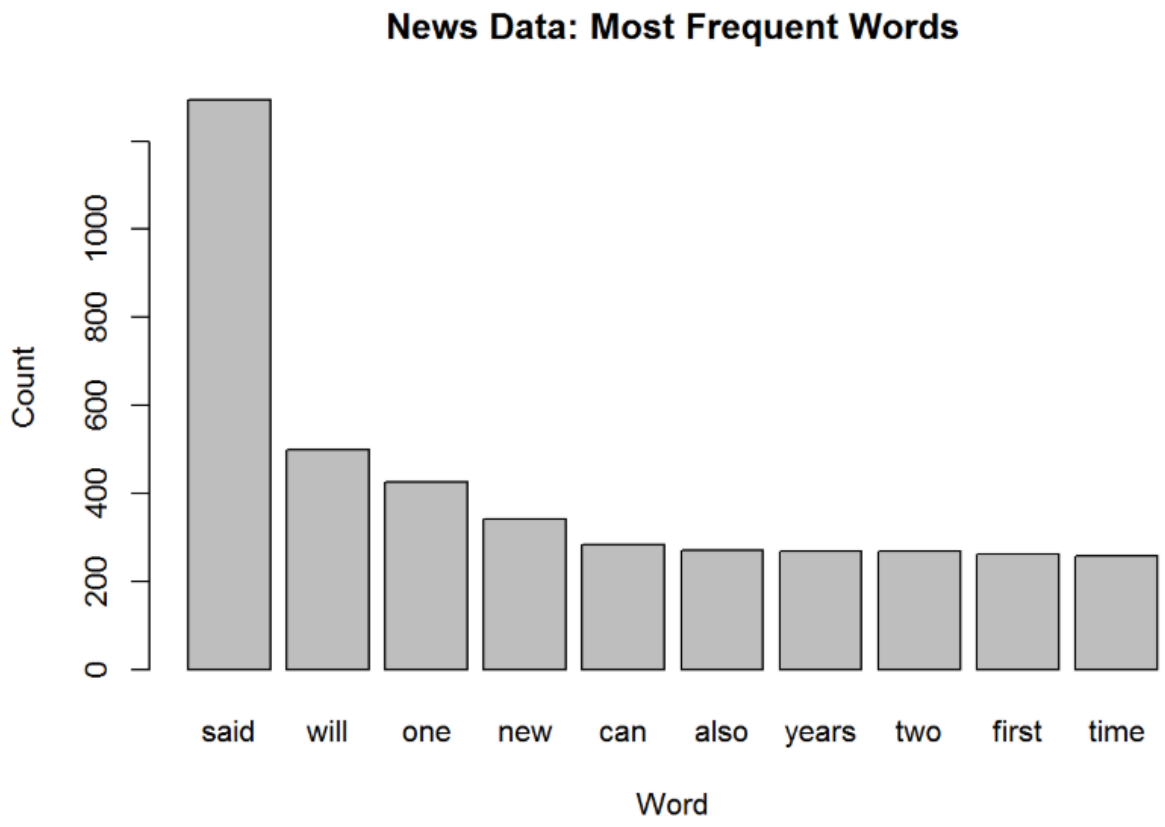


News Data

```
# Create corpus
corpus2 <- Corpus(VectorSource(news_sample))
```

```
# To lower case
corpus2 <- tm_map(corpus2, content_transformer(tolower))
# Remove punctuation marks
corpus2 <- tm_map(corpus2, removePunctuation)
# Remove numbers
corpus2 <- tm_map(corpus2, removeNumbers)
#remove stop words
corpus2 <- tm_map(corpus2, removeWords, stopwords("english"))
#Remove whitespaces
corpus2 <- tm_map(corpus2, stripWhitespace)
```

```
frequentWords <- head(sort(rowSums(as.matrix(TermDocumentMatrix(corpus2))),de
creasing=TRUE), 10)


barplot(frequentWords,
        main = "News Data: Most Frequent Words",
        xlab="Word",
        ylab = "Count")
```




```
term.doc.matrix2 <- TermDocumentMatrix(corpus2)
term.doc.matrix2 <- as.matrix(term.doc.matrix2)
word.freqs2 <- sort(rowSums(term.doc.matrix2), decreasing=TRUE)
dm2 <- data.frame(word=names(word.freqs2), freq=word.freqs2)
```

Most common words in the corpus

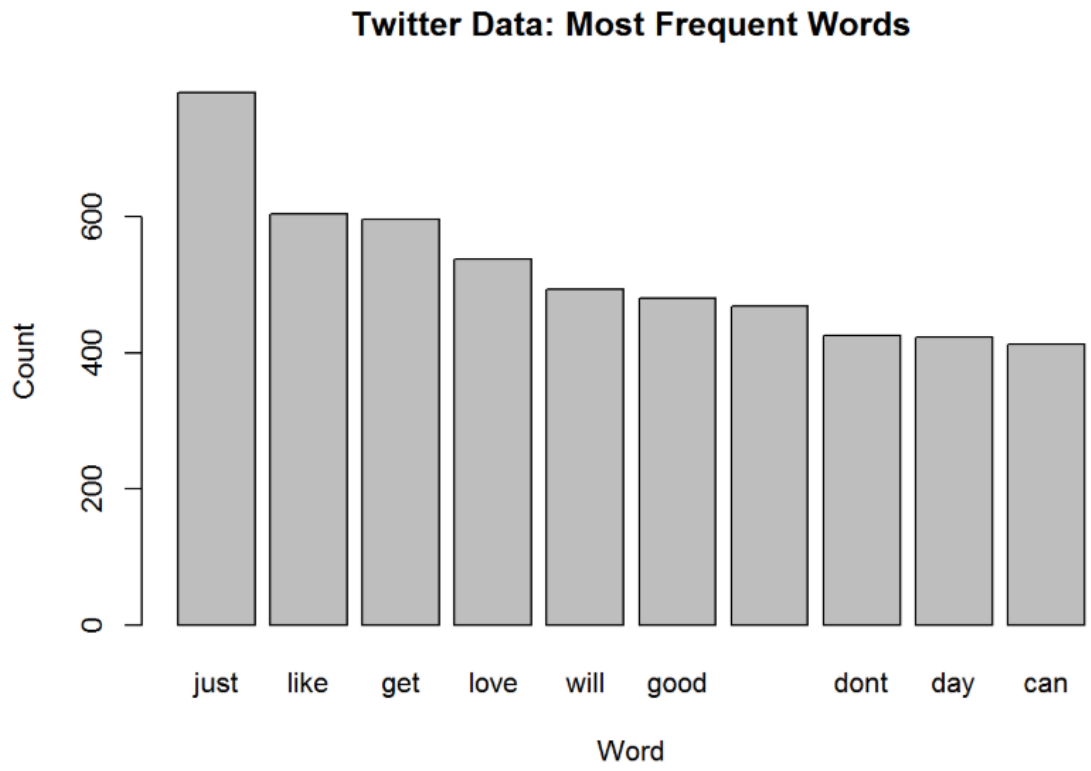
```
wordcloud(dm2$word, dm2$freq, min.freq= 100, random.order=TRUE, rot.per=.25,
colors=brewer.pal(8, "Dark2"))
```

Twitter Data

```
# Create corpus
corpus3 <- Corpus(VectorSource(twitter_sample))

## Convert Character Vector between Encodings
corpus3 <- tm_map(corpus3, content_transformer(function(x)
  iconv(x, to = "UTF-8", sub = "byte")))



# To lower case
corpus3 <- tm_map(corpus3, content_transformer(tolower))
# Remove punctuation marks
corpus3 <- tm_map(corpus3, removePunctuation)
# Remove numbers
corpus3 <- tm_map(corpus3, removeNumbers)
#remove stop words
```

```
corpus3 <- tm_map(corpus3, removeWords, stopwords("english"))
#Remove whitespaces
corpus3 <- tm_map(corpus3, stripWhitespace)
```

```
frequentWords <- head(sort(rowSums(as.matrix(TermDocumentMatrix(corpus3))),de
creasing=TRUE), 10)


barplot(frequentWords,
        main = "Twitter Data: Most Frequent Words",
        xlab="Word",
        ylab = "Count")
```




```
term.doc.matrix3 <- TermDocumentMatrix(corpus3)
term.doc.matrix3 <- as.matrix(term.doc.matrix3)
word.freqs3 <- sort(rowSums(term.doc.matrix3), decreasing=TRUE)
dm3 <- data.frame(word=names(word.freqs3), freq=word.freqs3)
```

```
wordcloud(dm3$word, dm3$freq, min.freq= 100, random.order=FALSE, rot.per=.25,
colors=brewer.pal(8, "Dark2"))
```

Future Analysis/Plans:

More models - N grams: bigrams, trigrams. Create a prediction model