Capstone Project - Milestone Report

Fernanda H. P. Gomes

October 31, 2018

Summary

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

1- Data loading

Below we will load into R the three files we will use in the prediction algorithm: blogs, news and twitter.

```
#Selecting the folder
setwd("~/Desktop/ Data Science/Coursera Capstone/data")

#loading the necessary packages
library(stringi)
library(tm)
```

```
## Loading required package: NLP
```

```
library(RWeka)
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
## The following object is masked from 'package:NLP':
##
##     annotate
```

```
library(quanteda)
```

```
## Package version: 1.3.14
```

```
## Parallel computing: 2 of 4 threads used.
```

```
## See https://quanteda.io for tutorials and examples.
```

```
##
## Attaching package: 'quanteda'
```

```
## The following objects are masked from 'package:tm':
##
##     as.DocumentTermMatrix, stopwords
```

```
## The following object is masked from 'package:utils':
##
##     View
```

```
#let's check how large is the data

cat("en_US.blogs.txt: " , file.info("./final/en_US/en_US.blogs.txt")$size / (1024*102
4) ,"mb")
```

```
## en_US.blogs.txt:  200.4242 mb
```

```
cat("en_US.news.txt: " , file.info("./final/en_US/en_US.news.txt")$size / (1024*1024)
 ,"mb")
```

```
## en_US.news.txt:  196.2775 mb
```

```
cat("en_US.twitter.txt: " ,file.info("./final/en_US/en_US.twitter.txt")$size / (1024*
1024) ,"mb")
```

```
## en_US.twitter.txt:  159.3641 mb
```

```
#importing the data
blogs<-file("./final/en_US/en_US.blogs.txt","r")
blogs_lines<-readLines(blogs, encoding = "UTF-8",skipNul = TRUE)

news<-file("./final/en_US/en_US.news.txt","r")
news_lines<-readLines(news, encoding = "UTF-8",skipNul = TRUE)

twitter<-file("./final/en_US/en_US.twitter.txt","r")
twitter_lines<-readLines(twitter, encoding = "UTF-8",skipNul = TRUE)
```

2- Data Preparation In order to plan the next steps we will start by understanding the available data and preparing that for some exploratory analysis

```
#first let's get a better understanding of the data that we have
DataStats <- rbind(stri_stats_general(news_lines), stri_stats_general(blogs_lines), s
tri_stats_general(twitter_lines))
DataStats <- as.data.frame(DataStats)
row.names(DataStats) <- c("news", "blogs", "twitter")
DataStats
```

```
##           Lines LinesNEmpty     Chars CharsNWhite
## news    1010242     1010242 203223154   169860866
## blogs    899288      899288 206824382   170389539
## twitter 2360148     2360148 162096241   134082806
```

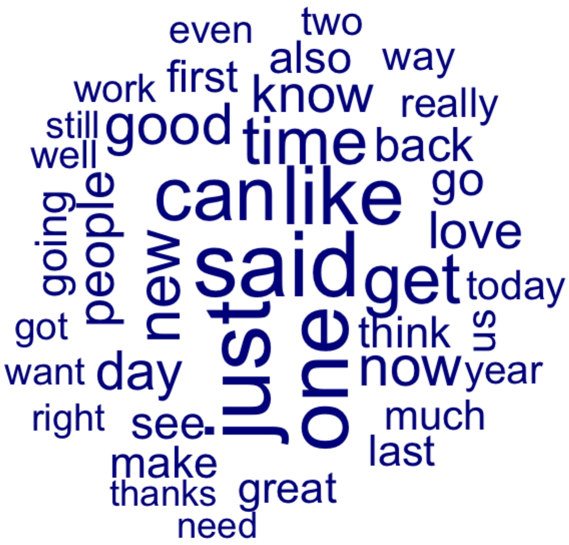
```
# as we have a lot of data, let's make a sample to help with exploratory analysis
set.seed(100)
blogs.sample <- sample(blogs_lines, length(blogs_lines) * 0.03)
news.sample <- sample(news_lines, length(news_lines) * 0.03)
lines_sample <- sample(twitter_lines, length(twitter_lines) * 0.03)

corpus.blog <- corpus(blogs.sample) #creating corpus
corpus.news <- corpus(news.sample) #creating corpus
corpus.lines <- corpus(lines_sample) #creating corpus

corpus <- corpus.blog + corpus.news + corpus.lines

summary(corpus)
```

```
## Corpus consisting of 128089 documents, showing 100 documents:
##
##     Text Types Tokens Sentences
##    text1    60     86         7
##    text2     7      7         1
##    text3     7      7         1
##    text4     5      5         2
##    text5     4      4         1
##    text6    29     31         3
##    text7    21     24         2
##    text8    99    168         7
##    text9    39     52         3
##   text10    25     26         1
##   text11    46     64         5
##   text12    12     13         1
##   text13    77    100         4
##   text14    55     82         2
##   text15   142    229        13
##   text16     5      5         1
##   text17    45     56         2
##   text18    69    102         6
##   text19     4      4         1
##   text20    11     12         1
##   text21   122    198         7
##   text22    75    103         4
##   text23    47     72         3
##   text24    27     30         2
##   text25    19     21         1
##   text26    36     43         2
##   text27    17     17         1
##   text28    36     44         4
##   text29     7      7         1
##   text30    14     16         2
##   text31    91    143         8
##   text32    22     27         2
##   text33     6      6         1
##   text34    52     75         5
##   text35     9      9         1
##   text36     9     10         3
##   text37     8      8         1
##   text38    51     67         3
##   text39    63     83         3
##   text40    30     35         4
##   text41    16     18         1
##   text42     2      2         1
##   text43    33     46         4
##   text44    19     20         1
##   text45    12     13         1
##   text46    91    130         9
##   text47     7      7         1
##   text48    78    103         4
##   text49    24     26         2
##   text50    52     65         2
##   text51    45     51         4
##   text52    43     72         4
##   text53    50     79         1
##   text54    92    128         4
##   text55    10     10         1
##   text56    16     17         3
##   text57    13     13         1
##   text58    12     13         1
##   text59     4      4         1
##   text60    13     14         1
##   text61    41     52         3
##   text62    82    102         5
##   text63     6      6         1
##   text64    11     11         1
##   text65    10     11         2
##   text66     7      7         1
##   text67    51     67         5
##   text68    40     42         1
##   text69     1      1         1
##   text70    63    113         4
##   text71    16     18         2
##   text72     8      8         1
##   text73     4      4         1
##   text74     5      5         1
##   text75    17     21         1
##   text76     6      6         1
##   text77     7      7         1
##   text78    85    116         6
##   text79     3      3         1
##   text80     5      5         1
##   text81    39     51         1
##   text82    58     80         2
##   text83    29     38         2
##   text84    34     38         2
##   text85    10     10         1
##   text86    44     63         5
##   text87    50     57         2
##   text88   112    169         7
##   text89    30     31         1
##   text90    82    126         4
##   text91    39     52         1
##   text92    11     11         1
##   text93    89    156         5
##   text94    35     43         2
##   text95    35     47         2
##   text96    14     14         2
##   text97     3      3         2
##   text98    68    104         4
##   text99     4      4         1
##  text100    33     43         2
## 
## Source: Combination of corpuses corpus.blog + corpus.news and corpus.lines
## Created: Mon Dec 10 23:02:01 2018
## Notes:
```

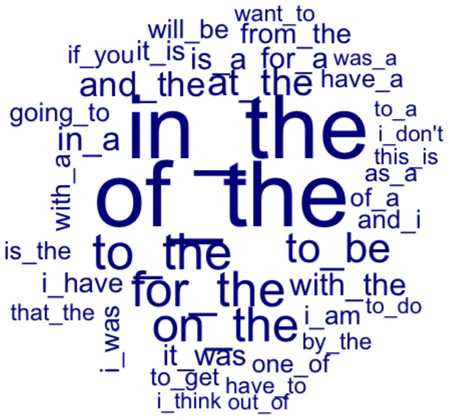
```
#creating tokens and cleaning the data

unigram <- tokens (corpus, remove_numbers = TRUE, remove_punct = TRUE, remove_separators = TRUE , ngrams =1 )

bigram <- tokens (corpus, remove_numbers = TRUE, remove_punct = TRUE, remove_separators = TRUE , ngrams =2 )

trigram <- tokens (corpus, remove_numbers = TRUE, remove_punct = TRUE, remove_separators = TRUE , ngrams =3 )

#creating the matrix
dfm.uni <- dfm (unigram, remove = stopwords("english"))
dfm.uni <- dfm_sort(dfm.uni) [,1:40]

dfm.bi <- dfm (bigram, remove = stopwords("english"))
dfm.bi <- dfm_sort(dfm.bi) [,1:40]

dfm.tri <- dfm (trigram, remove = stopwords("english"))
dfm.tri <- dfm_sort(dfm.tri) [,1:40]
```

3- Exploratory Analysis

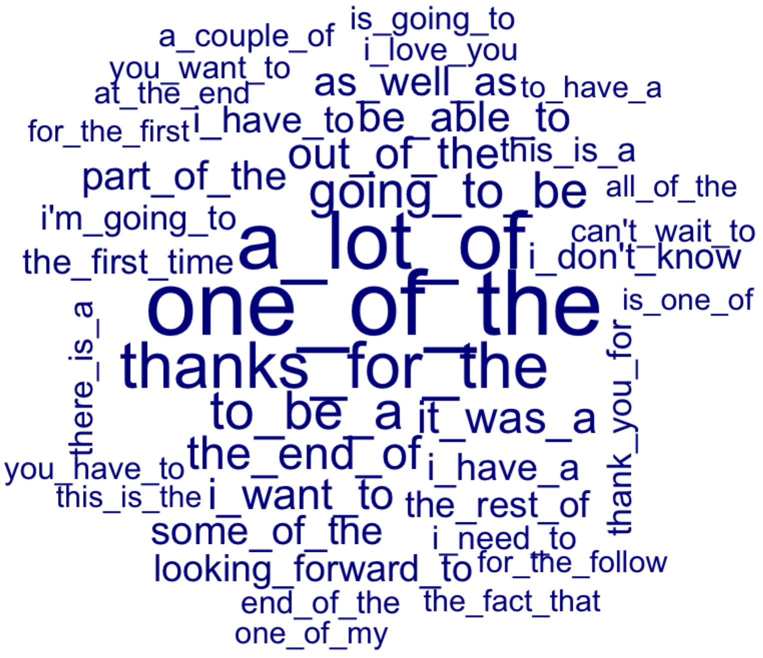
```
#Ploting top 40 by frequency

textplot_wordcloud(dfm.uni)
```



```
textplot_wordcloud(dfm.bi)
```



```
textplot_wordcloud(dfm.tri)
```



4 - Prediction algorithm

The goal is to create: - a prediction model based on the n-gram models biuld - a Shiny app (interface with the user): as the user enter words in a single textbox, the algorithm will be triggered on providing a list of suggesedd word that the user can select.

Challenge so fer: - the dataset is reallyy large, demanding a strategy on how better use the data without killing performance

Next steps: - work on further data cleanse - better define sample selection - build prediction algorithm - test and train datasets for a later prediction model