Data Science Capstone Milestone Report

Amlan Basu 28

September 2020

Introduction

The goal of this project is just to display that you've gotten used to working with the data and that you are on track to create your prediction algorithm. Please submit a report on R Pubs (http://rpubs.com/) that explains your exploratory analysis and your goals for the eventual app and algorithm. This document should be concise and explain only the major features of the data you have identified and briefly summarize your plans for creating the prediction algorithm and Shiny app in a way that would be understandable to a non-data scientist manager. You should make use of tables and plots to illustrate important summaries of the data set. The motivation for this project is to:

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.

```
library(plyr)
library(magrittr)
library(stringr)
library(stringi)
library(tm) library(RWeka)
library(SnowballC)
library(ggplot2)
```

Getting the Data

Download the dataset if it is not already there.

```
if(!file.exists("./data")){
dir.create("./data")
  url <-
"https://d396qusza40orc.cloudfront.net/dsscapstone/dataset/Courser a-
SwiftKey.zip"   download.file(Url, destfile="./data/Coursera-
SwiftKey.zip", mode = "wb")   unzip(zipfile="./data/Coursera-
SwiftKey.zip", exdir="./data")
}
```

Read the datasets

```
dataBlogs <- readLines("./data/en_US/en_US.blogs.txt", encoding = "UTF-8",
skipNul = TRUE)
dataNews <- readLines("./data/en_US/en_US.news.txt", encoding = "UTF-8", sk
ipNul = TRUE)
```

```
## Warning in readLines("./data/en_US/en_US.news.txt", encoding = "UTF-8",
:
## unvollstÃ¤ndige letzte Zeile in './data/en_US/en_US.news.txt' gefunden
```

```
dataTwitter <- readLines("./data/en_US/en_US.twitter.txt", encoding =
"UTF8", skipNul = TRUE)
```

Display statistics of the three datasets

```
stri_stats_general(dataBlogs)
```

```
##       Lines LinesNEmpty       Chars CharsNWhite
##      899288      899288   206824382   170389539
```

```
stri_stats_general(dataNews)
```

```
##       Lines LinesNEmpty       Chars CharsNWhite
##       77259       77259    15639408    13072698
```

```
stri_stats_general(dataTwitter)
```

```
##       Lines LinesNEmpty       Chars CharsNWhite
##     2360148     2360148   162096241   134082806
```

Data Preparation

Sample the data and create the corpus

```
subdataBlogs <- sample(dataBlogs, size = 1000) subdataNews
<- sample(dataNews, size = 1000) subdataTwitter <-
sample(dataTwitter, size = 1000) sampledData <-
c(subdataBlogs, subdataNews, subdataTwitter) corpus <-
VCorpus(VectorSource(sampledData))
```

Remove stopwords, punctuation, whitespaces, numbers etc. from the corpuses

```
toSpace <- content_transformer(function(x, pattern) gsub(pattern, " ",
x)) corpus <- tm_map(corpus, toSpace, "/|@|//|$|:|:)|*|&|!|?|_|-|#|")
corpus <- tm_map(corpus, content_transformer(tolower)) corpus <-
tm_map(corpus, removeNumbers) corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, removeWords, stopwords())
corpus <- tm_map(corpus, stemDocument) corpus <-
tm_map(corpus, stripWhitespace)
```

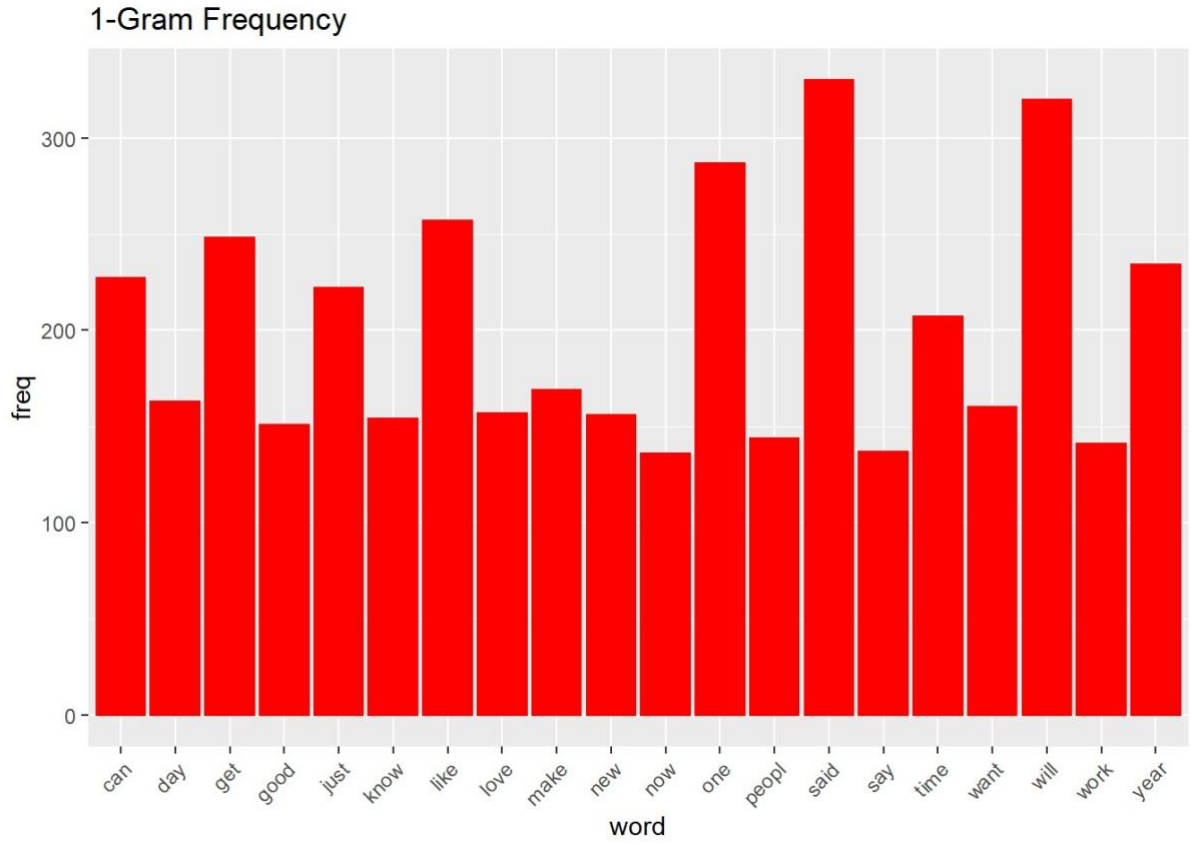
Create the DocumentTermMatrizes

```
dtm1 <- TermDocumentMatrix(corpus) bigram <- function(x)
NGramTokenizer(x, Weka_control(min = 2, max = 2)) dtm2 <-
TermDocumentMatrix(corpus, control = list(tokenize = bigram)) trigram <-
function(x) NGramTokenizer(x, Weka_control(min = 3, max = 3)) dtm3 <-
TermDocumentMatrix(corpus, control = list(tokenize = trigram))
```

Data Exploration

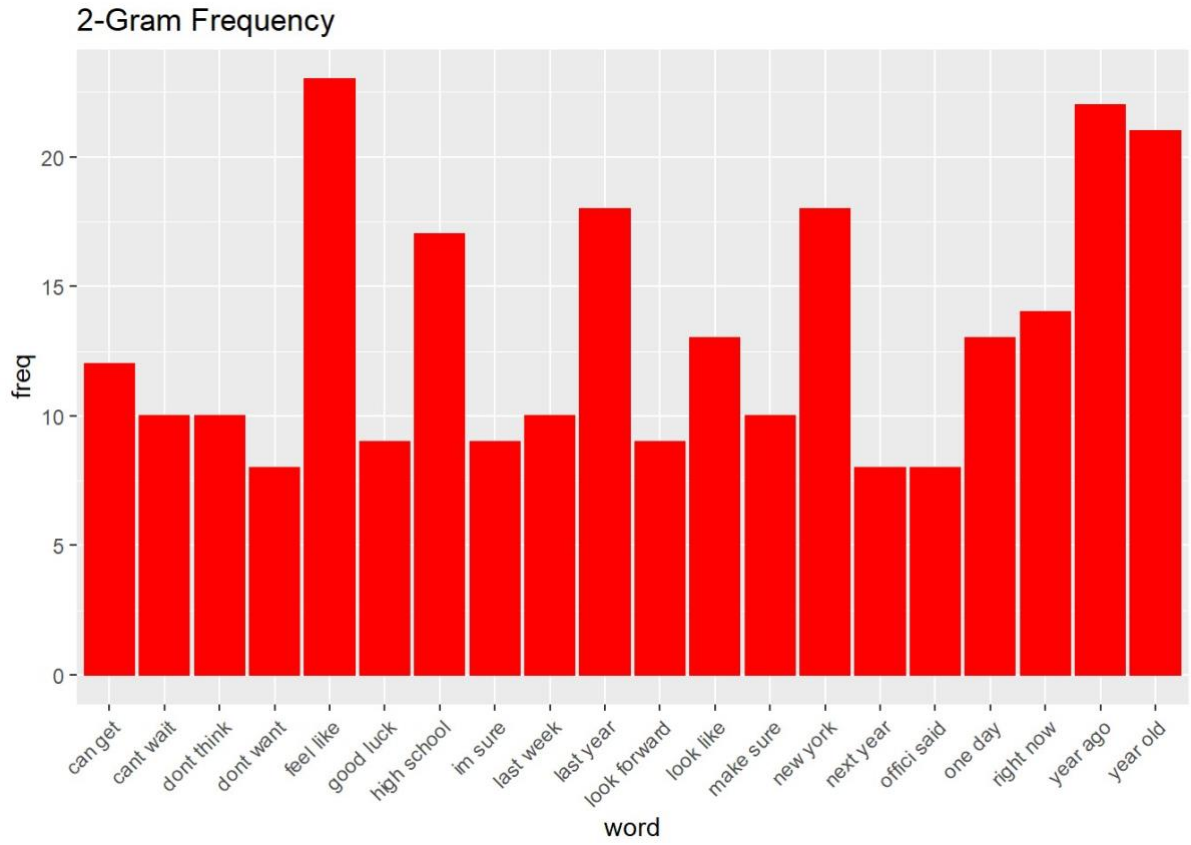
1-Gram Frequency

```
freq1 <- rowSums(as.matrix(dtm1)) freq1 <- sort(freq1,
decreasing = TRUE) dfFreq1 <- data.frame(word =
names(freq1), freq=freq1) ggplot(dfFreq1[1:20, ],
aes(word, freq)) +   geom_bar(stat="identity",
fill="red", colour="red") +
  theme(axis.text.x=element_text(angle=45, hjust=1)) + ggtitle("1-Gram Freq
uency")
```



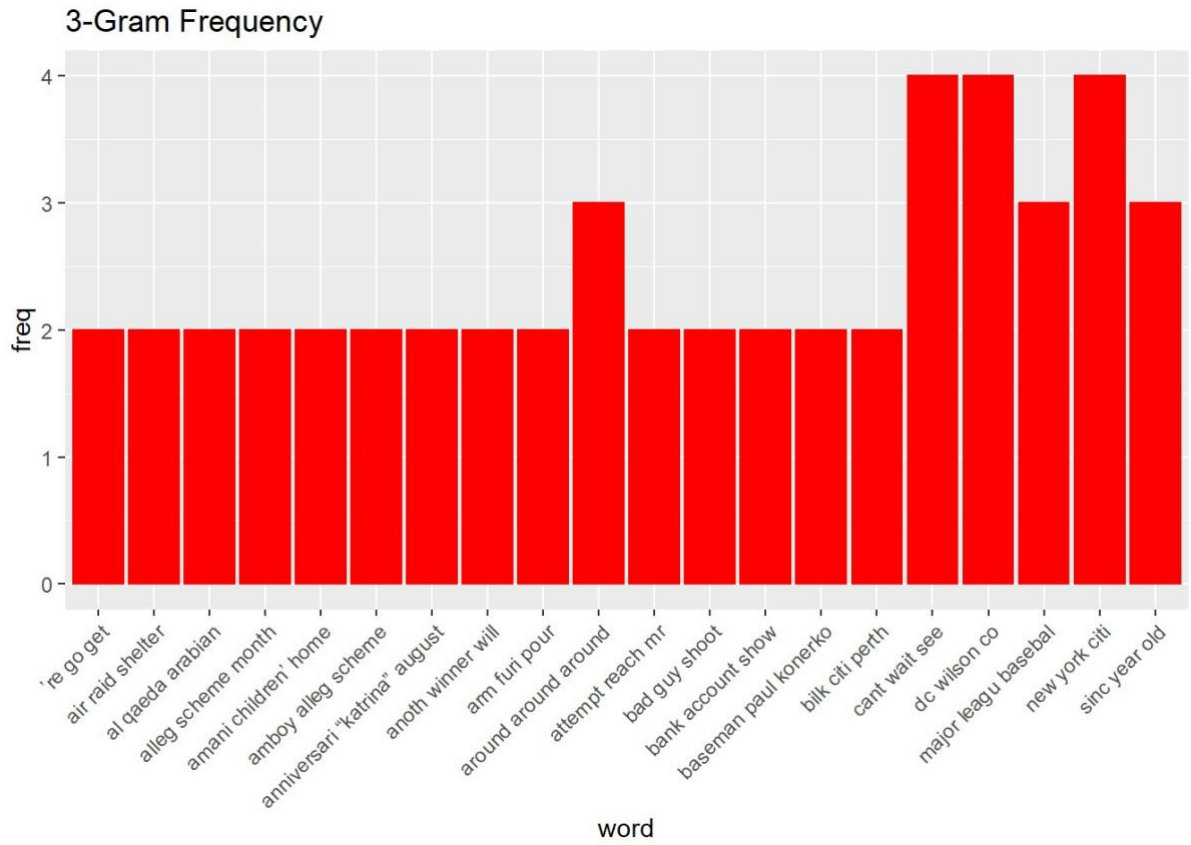
2-Gram Frequency

```
freq2 <- rowSums(as.matrix(dtm2)) freq2 <- sort(freq2,
decreasing = TRUE) dfFreq2 <- data.frame(word =
names(freq2), freq=freq2) ggplot(dfFreq2[1:20, ],
aes(word, freq)) +   geom_bar(stat="identity",
fill="red", colour="red") +
  theme(axis.text.x=element_text(angle=45, hjust=1)) + ggtitle("2-Gram Freq
uency")
```



3-Gram Frequency

```
freq3 <- rowSums(as.matrix(dtm3)) freq3 <- sort(freq3,
decreasing = TRUE) dfFreq3 <- data.frame(word =
names(freq3), freq=freq3) ggplot(dfFreq3[1:20, ],
aes(word, freq)) +   geom_bar(stat="identity",
fill="red", colour="red") +
  theme(axis.text.x=element_text(angle=45, hjust=1)) + ggtitle("3-Gram Freq
uency")
```



Future work

The goal is to create a predictive model which predicts the most probable words to follow an input from the user. This model will be evaluated and deployed as a shiny application.