Data Science - Capstone Milestone Report

WQ

This is a milestone report - prelude to the actual final product of a text prediction Shiny app.

Aims of this report

As highlighted in the “Milestone Report Rubic” There motivation of this report is to:

-Demonstrate that you’ve downloaded the data and have successfully loaded it in.

-Create a basic report of summary statistics about the data sets.

-Report any interesting findings that you amassed so far.

-Get feedback on your plans for creating a prediction algorithm and Shiny app.

Upload Dataset and Packages needed for the Report

```
#List of R Packages that will be used in this report.
library(tm)
library(stringi)
library(stringr)
library(RWeka)
library(SnowballC)
library(ggplot2)
library(wordcloud)

#Uploading the datasets
twitter <- readLines("C:/Users/User/Documents/R/final/en_US/en_US.twitter.txt", encodin
g="UTF-8")
twitter <- iconv(twitter, from="latin1", to="ASCII", sub="")

blogs <- readLines("C:/Users/User/Documents/R/final/en_US/en_US.blogs.txt", encoding="UT
F-8")
blogs <- iconv(blogs, from="latin1", to="ASCII", sub="")

news <- readLines("C:/Users/User/Documents/R/final/en_US/en_US.news.txt", encoding="UTF-
8")
news <- iconv(news, from="latin1", to="ASCII", sub="")
```

Brief Outlook of the Datasets

To understand the length, class and mode of the 3 datasets

```
summary(twitter)
```

```
##    Length     Class      Mode
##   2360148 character character
```

```
summary(blogs)
```

```
##    Length     Class      Mode
##    899288 character character
```

```
summary(news)
```

```
##    Length     Class      Mode
##     77259 character character
```

Size of the files

```
paste(file.info("C:/Users/User/Documents/R/final/en_US/en_US.twitter.txt")$size / (1024*1
024), 'MB')
```

```
## [1] "159.364068984985 MB"
```

```
paste(file.info("C:/Users/User/Documents/R/final/en_US/en_US.blogs.txt")$size / (1024*102
4), 'MB')
```

```
## [1] "200.424207687378 MB"
```

```
paste(file.info("C:/Users/User/Documents/R/final/en_US/en_US.news.txt")$size / (1024*102
4), 'MB')
```

```
## [1] "196.277512550354 MB"
```

The main takeaway from this set of data is that the 3 files are generally big, about 550MB in total. This means if I were to analyze the entire datasets, it will take up considerable time and computer memory.

Cleaning and Combing of Datasets

In the next step, we limit the amount of data we use from the 3 files and combing them as one. We will then proceed to clean the file to remove irrelevant words.

```
#Combining the sampled data
Tsam <- twitter[sample(1:length(twitter), 50000)]
Nsam <- news[sample(1:length(news), 50000)]
Bsam <- blogs[sample(1:length(blogs), 50000)]
Combinedsam <- c(Tsam, Nsam, Bsam)

#Cleaning the sampled data
Corpus <- Corpus(VectorSource(list(Combinedsam)))
Corpus <- tm_map(Corpus, content_transformer(tolower))
Corpus <- tm_map(Corpus, content_transformer(removePunctuation))
Corpus <- tm_map(Corpus, content_transformer(removeNumbers))
Corpus <- tm_map(Corpus, removeWords, stopwords("en"))
Corpus <- tm_map(Corpus, stripWhitespace)
Corpus <- tm_map(Corpus, stemDocument, language="en")
```

We proceed to make a wordcloud to give a rough idea what we will be expecting in our analysis next.

```
wordcloud(Corpus, scale=c(5,0.75), max.words=150, random.order=FALSE, rot.per=0.25, us
e.r.layout=FALSE, colors=brewer.pal(8, "Dark2"))
```

As we can see that the three main words are “said”, “will” and “one”.

Analysis

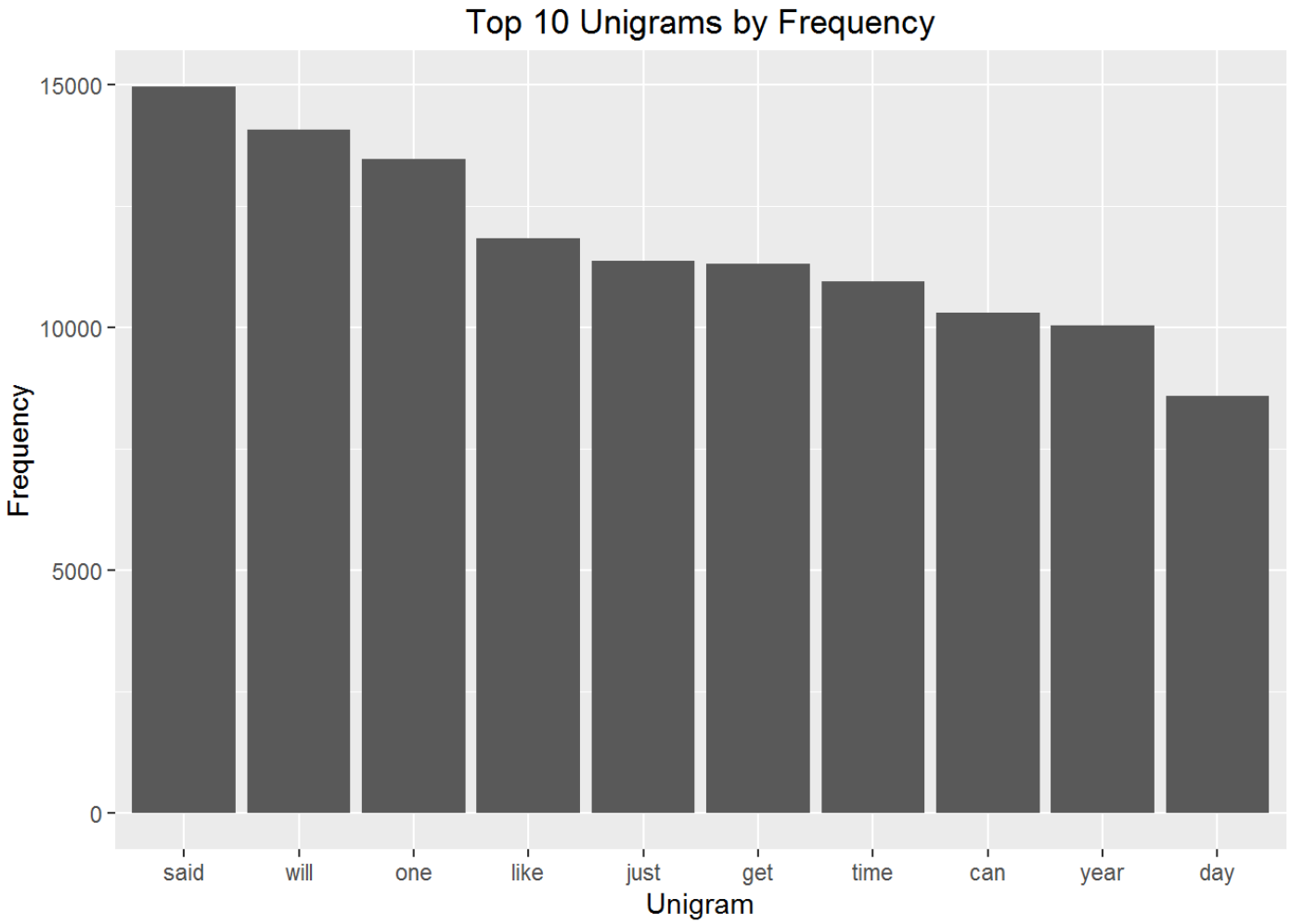
For the analysis of the dataset, we will proceed to carry out Unigram, Bigram and Trigram which make used of the RWeka Package mainly. ps: I’ve encountered some issue running the codes initially and utlimately realized it was a problem with the Java running on my computer. Do download Java beforehand.

Unigram

```
unizer <- function(x) NGramTokenizer(x, Weka_control(min=1, max=1))
unigram <- DocumentTermMatrix(Corpus, control=list(tokenize=unizer))

uniDF <- data.frame(Term=unigram$dimnames$Terms,
                        Freq=unigram$v)
uniDF <- uniDF[with(uniDF, order(-Freq)), ]

ggplot(head(uniDF, 10), aes(reorder(Term, -Freq), Freq)) +
    geom_bar(stat="identity") +
    ggtitle("Top 10 Unigrams by Frequency") +
    xlab("Unigram") + ylab("Frequency")
```



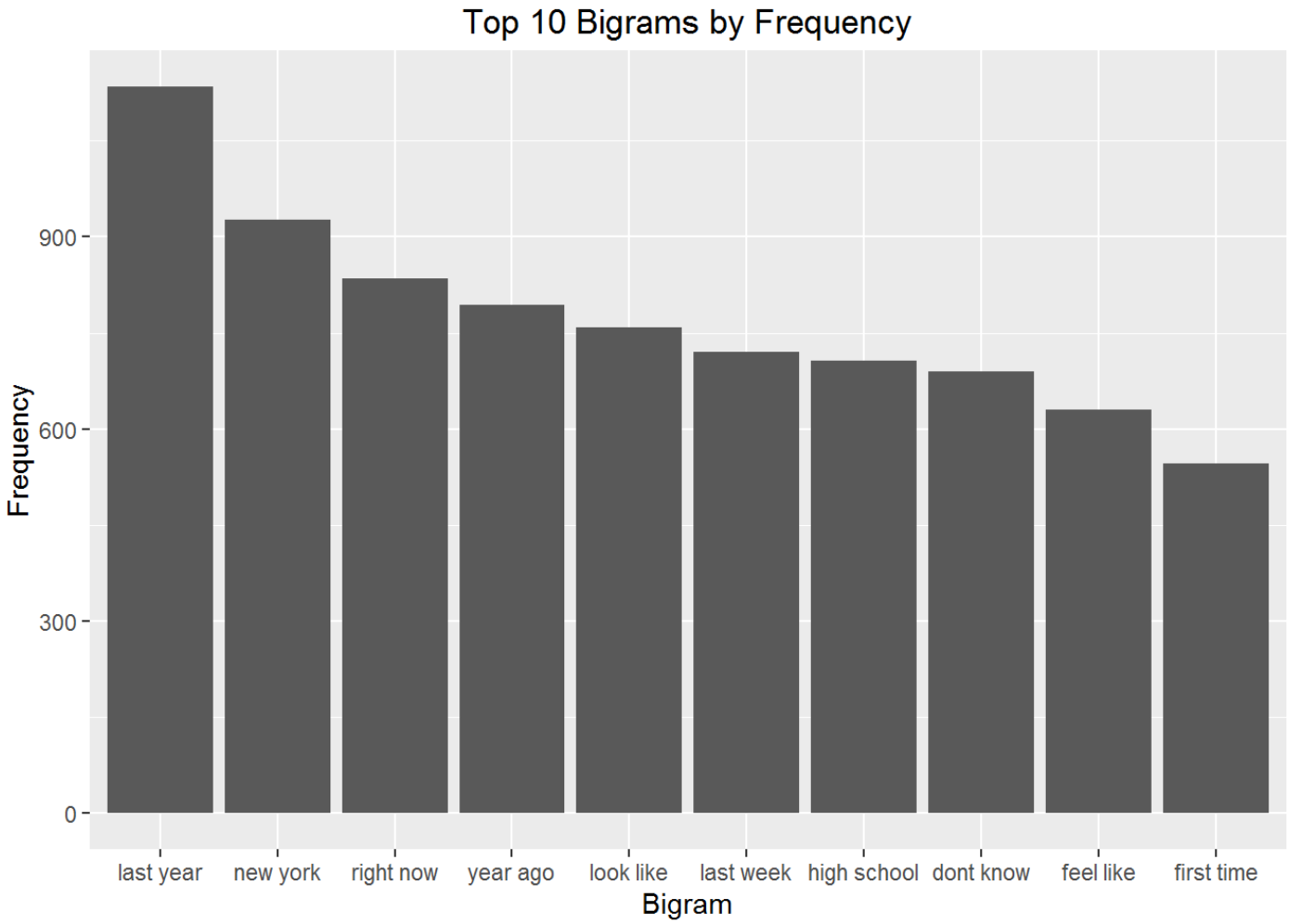
The top Unigrams are “said” followed by “will” and “one”.

Bigram

```
bizer <- function(x) NGramTokenizer(x, Weka_control(min=2, max=2))
bigram <- DocumentTermMatrix(Corpus, control=list(tokenize=bizer))

biDF <- data.frame(Term=bigram$dimnames$Terms,
                        Freq=bigram$v)
biDF <- biDF[with(biDF, order(-Freq)), ]

ggplot(head(biDF, 10), aes(reorder(Term, -Freq), Freq)) +
    geom_bar(stat="identity") +
    ggtitle("Top 10 Bigrams by Frequency") +
    xlab("Bigram") + ylab("Frequency")
```



The top bigrams are “last year”, followed by “new york” and “right now”.

Trigram

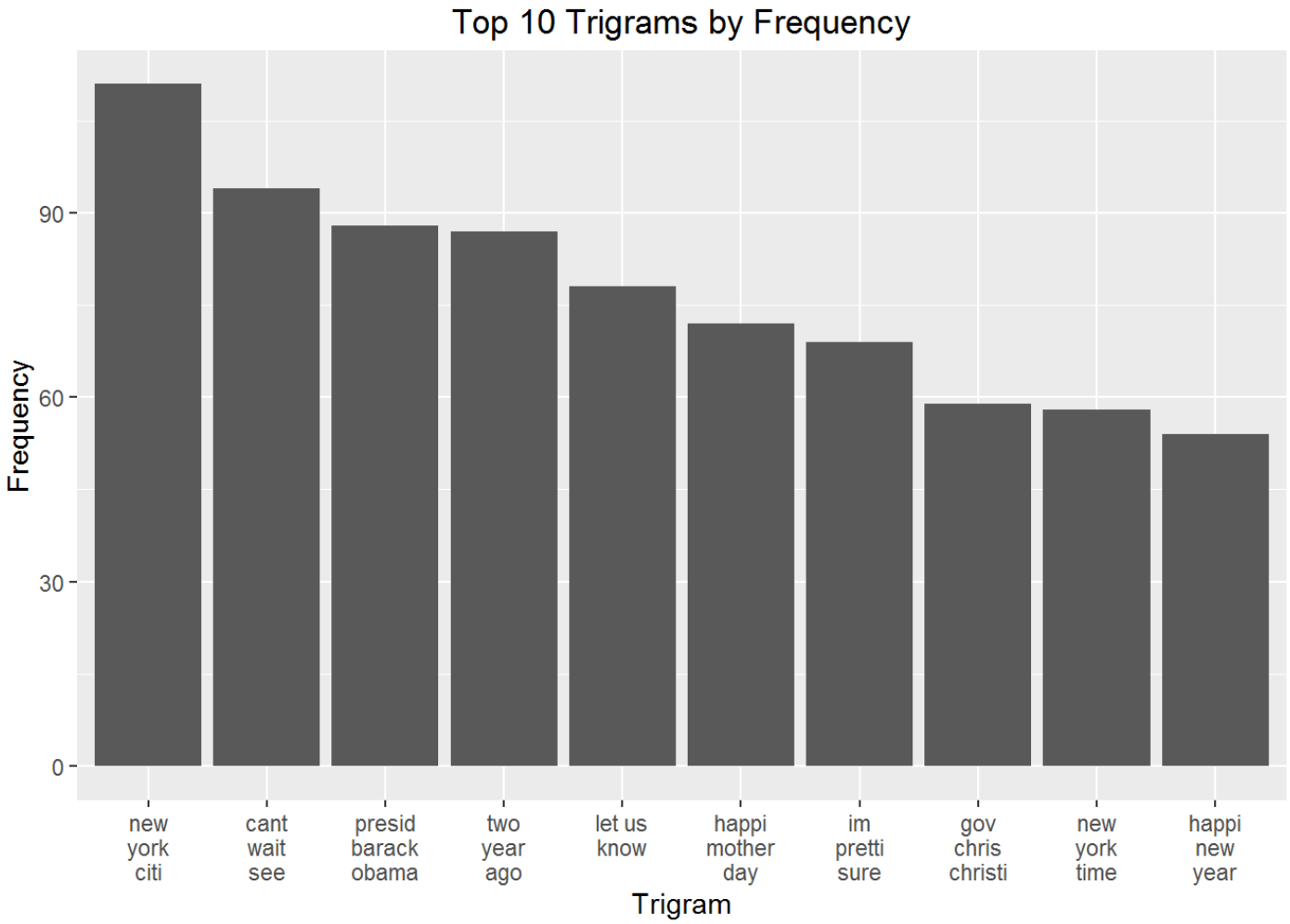
```
trizer <- function(x) NGramTokenizer(x, Weka_control(min=3, max=3))
trigram <- DocumentTermMatrix(Corpus, control=list(tokenize=trizer))

triDF <- data.frame(Term=trigram$dimnames$Terms,
                    Freq=trigram$v)
triDF <- triDF[with(triDF, order(-Freq)), ]

triDF10 <- head(triDF, 10)

triDF10$newTerm = str_wrap(triDF10$Term, width = 6)

ggplot(head(triDF10, 10), aes(reorder(newTerm, -Freq), Freq)) +
    geom_bar(stat="identity") +
    ggtitle("Top 10 Trigrams by Frequency") +
    xlab("Trigram") + ylab("Frequency")
```



The top trigrams are “new york citi”, followed by “cant wait see” and “presid barack obama”.

Conclusion

In the milestone report, it allows me to have a better idea about the dataset we will be dealing with. However for the final project - creating a predictive word ShinyApp will be a totally different attempt.Despite limited the dataset to 150,000, the amount of time needed to run the codes is still considerable long. Hence I will expect even more time and resources are needed for doing up the Shinyapp.