Title : "Peer-graded assignment Milestone Report"

author: "*Swati*"
date: '2022-04-30'
output: html_document

Introduction

In this Data Science Capstone, the goal is to predict what is written next based on the last few words that are just typed.

This report explains the exploratory analysis and some modelling for the eventual app and algorithm.

Downloading and reading in files

```
<!-- setwd("~/Downloads/capstone_project_week.1/Coursera-SwiftKey/final/en_US") -->
destfile = "./Coursera-SwiftKey.zip"
if(!file.exists(destfile)){
  url = "https://d396qusza40orc.cloudfront.net/dsscapstone/dataset/Coursera-SwiftKey.zip"
  file <- basename(url)
  download.file(url, file, method="curl")
  unzip(file)
}
news <- readLines("final/en_US/en_US.news.txt", encoding = 'UTF-8',warn = FALSE)
twitter <- readLines("final/en_US/en_US.twitter.txt", encoding = 'UTF-8',warn = FALSE)
blogs <- readLines("final/en_US/en_US.blogs.txt", encoding = 'UTF-8',warn = FALSE)
```

Exploratory data analysis wordcounts and linecounts

```
library(ngram)
line_news<-length(news)
line_twitter<-length(twitter)
line_blogs<-length(blogs)

wc_news<-wordcount(news)
wc_twitter<-wordcount(twitter)
wc_blogs<-wordcount(blogs)

a<-rbind(line_news,line_twitter,line_blogs)
b<-rbind(wc_news,wc_twitter,wc_blogs)
c<-as.data.frame(cbind(a,b))
names(c)<-c("nr of lines","nr of words")
rownames(c)<-c("news","twitter","blogs")
c
```

```
##  nr of lines nr of words
## news       1010242    34372530
## twitter    2360148    30373543
## blogs       899288    37334131
```

Files are too large to process. Therefore 1% sample is taken of each, and the files are combined

```
library(RWeka)
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
```

```
set.seed(11000)
c_blogs <- sample(blogs, length(blogs)*0.01)
c_news <- sample(news, length(news)*0.01)
c_twitter <- sample(twitter, length(twitter)*0.01)
c_combi=c(c_blogs,c_news,c_twitter)
```

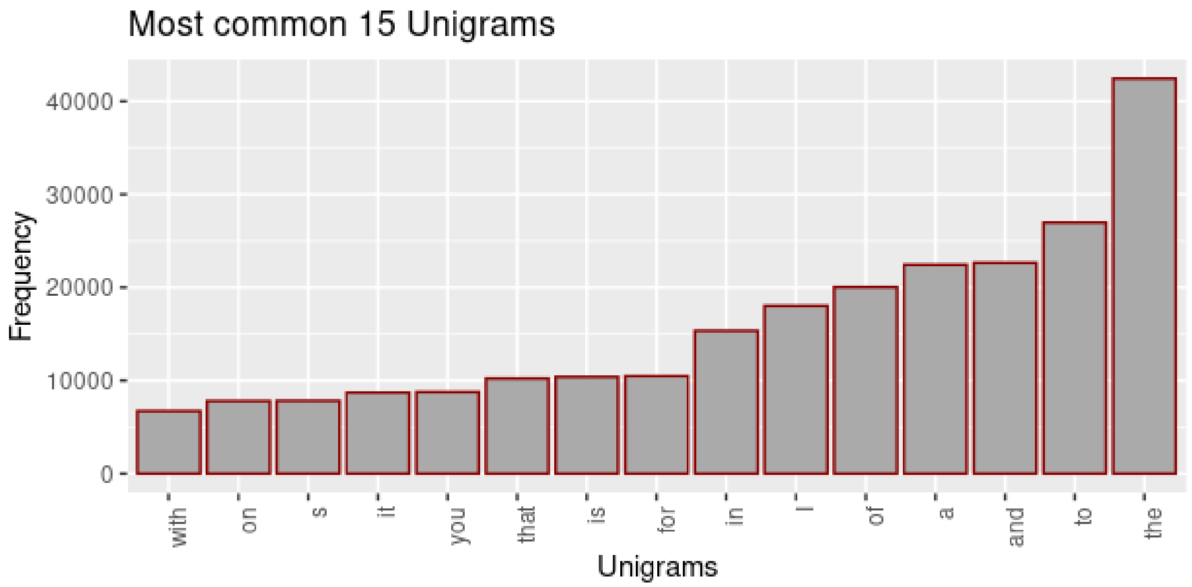
1-, 2- and 3- ngrams and plots

```
unigram_combi <- NGramTokenizer(c_combi, Weka_control(min = 1, max = 1))
bigram_combi <- NGramTokenizer(c_combi, Weka_control(min = 2, max = 2))
trigram_combi <- NGramTokenizer(c_combi, Weka_control(min = 3, max = 3))

unigram_combi<-data.frame(table(unigram_combi))%>%arrange(desc(Freq))
bigram_combi<-data.frame(table(bigram_combi))%>%arrange(desc(Freq))
trigram_combi<-data.frame(table(trigram_combi))%>%arrange(desc(Freq))

df_ngram<-as.data.frame(cbind(unigram_combi[1:15,],bigram_combi[1:15,],trigram_combi
[1:15,]))
names(df_ngram)[c(2,4,6)]<-c("Freq1","Freq2","Freq3")
df_ngram
```

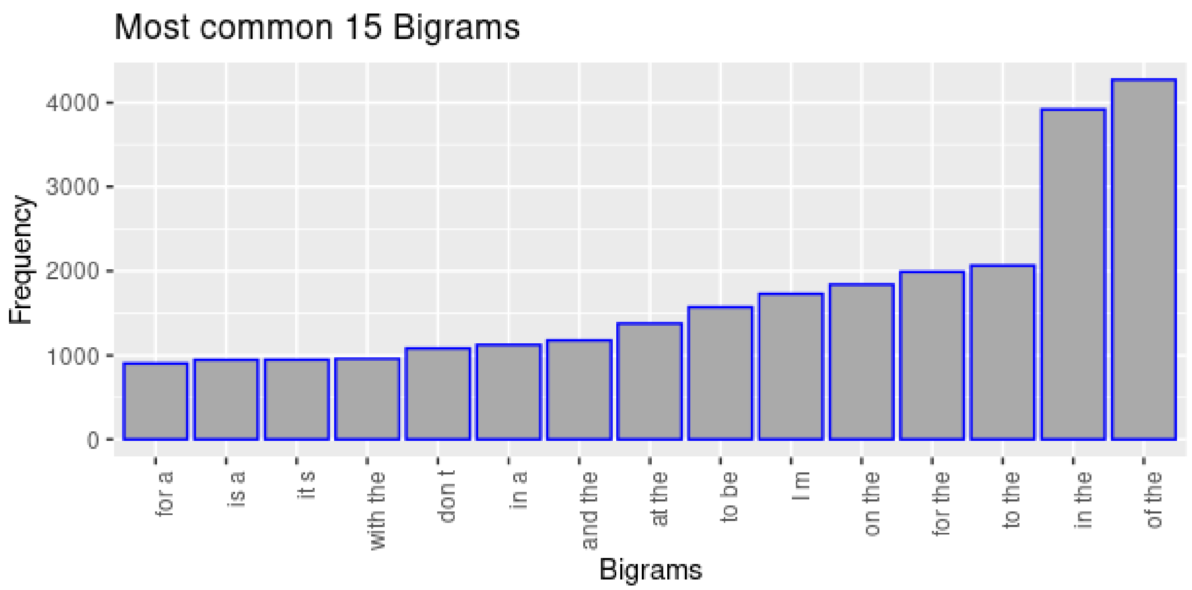
```
##    unigram_combi Freq1 bigram_combi Freq2  trigram_combi Freq3
## 1            the 42437       of the  4272        I don t   413
## 2           to 26949      in the  3916     one of the   285
## 3          and 22624       to the  2063       a lot of   258
## 4            a 22421     for the  1990        I can t   221
## 5           of 20040      on the  1838        to be a   189
## 6            I 18017        I m  1730        I m not   188
## 7           in 15362       to be  1570 Thanks for the   169
## 8          for 10469      at the  1375     be able to   159
## 9           is 10377     and the  1175    going to be   154
## 10        that 10210        in a  1122     the end of   154
## 11         you  8788       don t  1081      I want to   146
## 12          it  8724    with the   958    don t know   139
## 13           s  7825        it s   946    as well as   138
## 14          on  7800        is a   944       the U S   132
## 15        with  6753       for a   901       I didn t   128
```

Plots

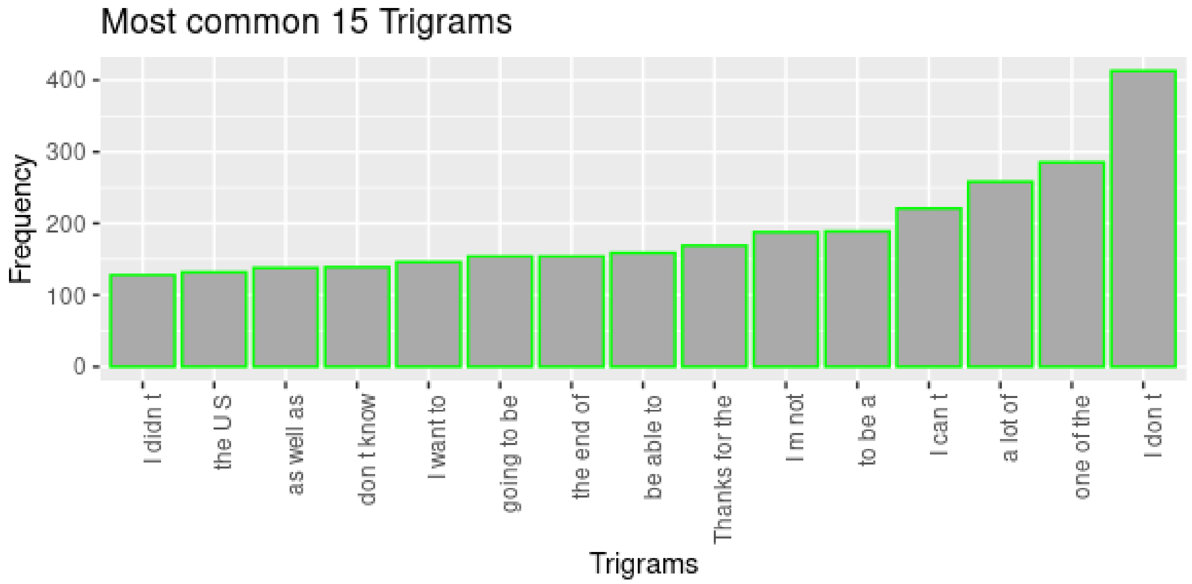
```
library(ggplot2)
ggplot(df_ngram, aes(x=reorder(unigram_combi,Freq1), y=(Freq1))) +
  geom_bar(stat="Identity", fill="#AAAAAA",color="darkred")+
  xlab("Unigrams") + ylab("Frequency")+
  ggtitle("Most common 15 Unigrams")+
  theme(axis.text.x=element_text(angle=90, hjust=1))
```



```
ggplot(df_ngram, aes(x=reorder(bigram_combi,Freq2), y=(Freq2))) +
  geom_bar(stat="Identity", fill="#AAAAAA", color="blue")+
  xlab("Bigrams") + ylab("Frequency")+
  ggtitle("Most common 15 Bigrams")+
  theme(axis.text.x=element_text(angle=90, hjust=1))
```



```
ggplot(df_ngram, aes(x=reorder(trigram_combi,Freq3), y=(Freq3))) +
  geom_bar(stat="Identity", fill="#AAAAAA", color="green")+
  xlab("Trigrams") + ylab("Frequency")+
  ggtitle("Most common 15 Trigrams")+
  theme(axis.text.x=element_text(angle=90, hjust=1))
```



Summary and conclusion

We have done examining the dataset and get some intereting findings from the exploratory analysis. Now we are ready to train and create our first predictive model. Machine Learning is an iterative process where we preprocess the training data, then train and evaluate the model and repeat the steps again iteratively to get better performace model based on our evaluation metrics.

Before we end this report, It is important to note that each of the steps are important and each steps need to be re-evaluated continuosly to get really working and accurate ML model for our predictive text app. We are looking forward on the next report on the predictive model and shiny app we'll going to build!