Taisong_Jing_Milestone_Report

Taisong Jing

Saturday, November 15, 2014

This is a milestone report on the capstone project towards specialization in data science. The goal of this natural language processing project is to build a word prediction model by text mining. The three data sets "en_US.blogs.txt", "en_US.news.txt", and "en_US.twitter.txt" are acquired from blogs, news, and twitter. This report is a first exploratory analysis of the texts in the provided data sets.

1. The original data set and subsetting

The basic summary statistics of the three data sets are as follows (counted using Python)):

```
##                      lines    words
## en_US.blog.txt      899288 37334131
## en_US.news.txt     1010242 34372531
## en_US.twitter.txt 2360148 30373583
```

Since the data sets are fairly large the computation resources are limited, we only take about 10% sample of the data set to analyze. At the same time, we discard all the characters that are not alpha-numeric, and replace the "!", and "?" symbles with ".". Here is the subsetting R code for the "en_US.twitter.txt"; the subsetting for the other two files are the same. The three subsets of the files are stored in "blogsSample.txt", "newsSample.txt", and "twitterSample.txt".

```
##take sample subset of the files
con<-file("en_US.twitter.txt","r")
readSize<-100
set.seed(100)
chunk<-character(0)
write(chunk, file="twitterSample.txt")
chunk<-readLines(con, readSize)
while ( length(chunk) > 0 ) {
    if (runif(1)<0.1) {
        chunk<-gsub("[^(a-zA-Z0-9 |\\.|\\!|\\?) ]", "", chunk)
        chunk<-gsub("[\\.|\\!|\\?]", " . ", chunk)
        write(chunk, file="twitterSample.txt", append=TRUE)
    }
    chunk<-readLines(con, readSize)
}
close(con)
```

The basic statistics of the three sample files are:

```
##                      lines    words
## blogsSample.txt     899288 37334131
## newsSample.txt     1010242 34372531
## twitterSample.txt 2360148 30373583
```

2. Most frequent words and phrases in the data sets

The frequencies of the words and two-word-phrases in the sample files are summarized as follows:

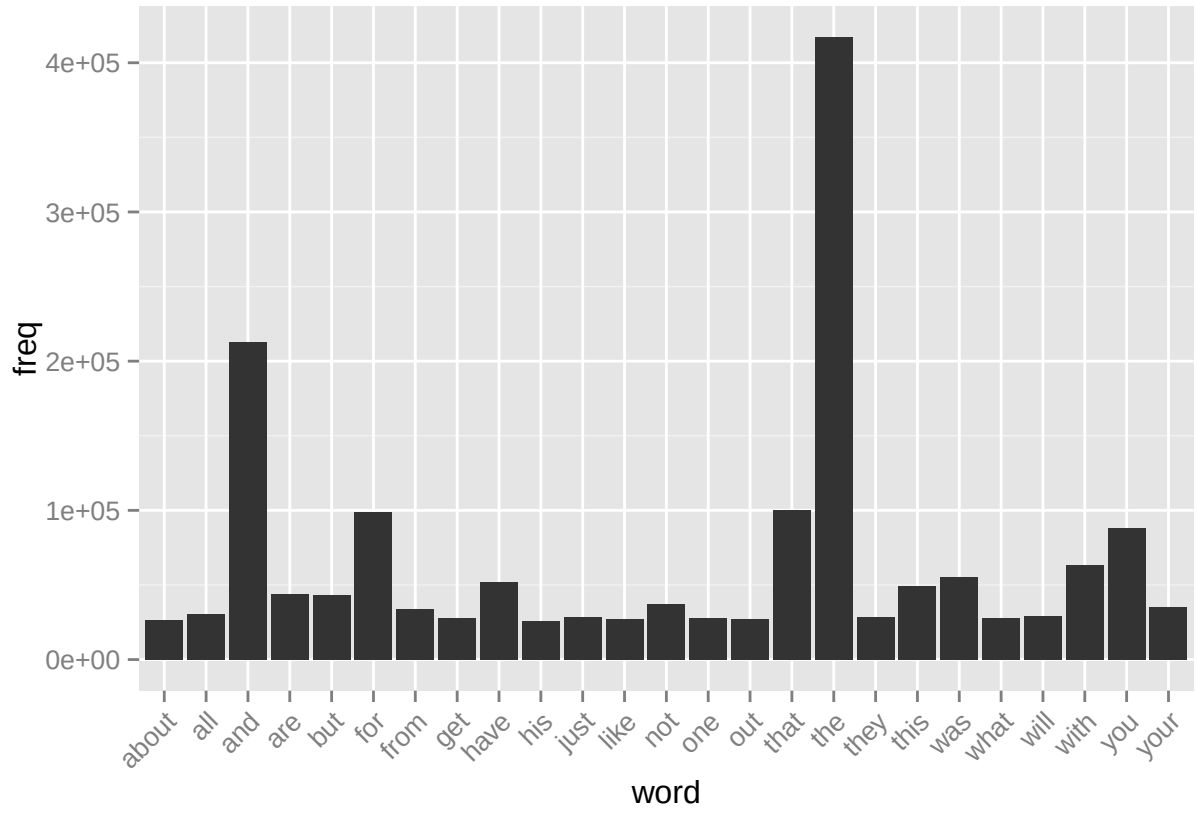
```
##                                          X90..instances
## words               5000 (6281689 counts out of 6952000)
## two-word phrases 1250000 (7685044 counts out of 8499338)
##                                       X50..instances
## words            225 (3470393 counts out of 6952000)
## two-word phrases 225 (4216136 counts out of 8499338)
```

The histograms of the most frequently used words and two-word-phrases are as follows:

- word:

```
## Loading required package: rJava
## Loading required package: xlsxjars
```

```
## Warning: package 'xlsxjars' was built under R version 3.1.2
```



3. Wordclouds for the three sample files

The wordcloud is a favorable way to visualize the high-frequency words in a piece of text. From the three wordclouds (please find the repository) we can see there are more high-frequency words in twitter than in blogs or news. A possible explanation for this phenomenon is that the language used on twitters are more casual and oral-based.

4. Plans on the prediction algorithm and the shiny app

The prediction algorithm will be based on the n-gram model. Due to the limitation of the computing resources, 2-gram or 3-gram would be a reasonable choice. To improve the performance, I will try some smoothing techniques such as Good-Turing smoothing.

My current idea on the shiny app is to let the user come up with a few words, give a number for each word to indicate its position in the sentence, then the app will fill out the sentence according to the prediction algorithm. For example, if the user chooses “I”, “bike” and “school” with position “1”, “4”, and “6”, then a possible prediction is “I ride a bike to school.” Sometimes the prediction sentence may be meaningful or funny!