Capstone Project Report

Yin Bian

Tuesday, November 29, 2015

Introduction:

This exploratory analysis looks at the interesting features discovered during the fall 2014 session of the Coursera Data Science capstone project. The goal of the capstone project is to build a Shiny app that provides next word text prediction based on user supplied text. The data used for this analysis was provided by Swiftkey and consisted of three data sets for various languages bases on tweets collected on Twitter, blog entries, and news sources[1]. This study concentrates on the US English data sets. All analysis was performed in R[2], and the report was generated using R-markdown[3], but the code is not displayed to increase readability.

Data Preprocessing:

Each data file was initially loaded into R to look at the dimensions of the data.

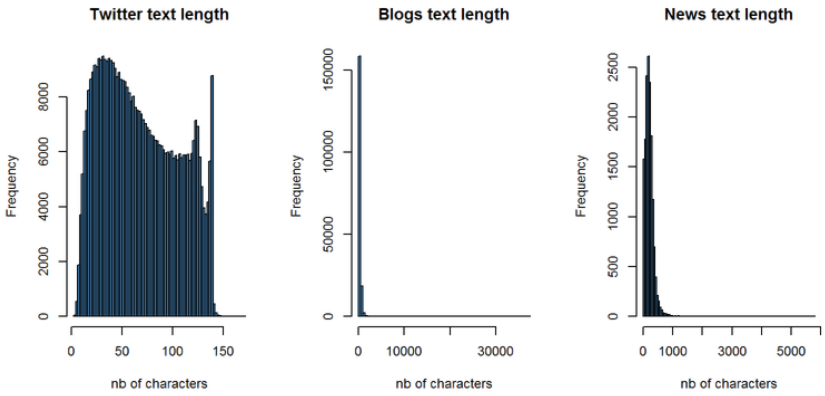
Each file was first loaded into R, and the number of lines and object size of the file were verified. As size of each data set was quite large [Table 1], it was decided to use 20% of the available data for further analysis.

Table 1 Basic statistics of text files

```
##          Length     Object Size
## twitter 2360148 316037344 bytes
## news      77259  20111392 bytes
## blogs    899288 260564320 bytes
```

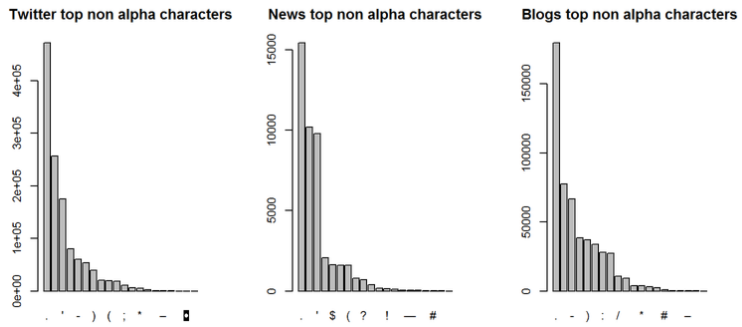
After loading in the data, it was found that the twitter sample had many more individual entries, as the original file sizes were similar it was decided to plot histograms of the lengths of the individual entries [Figure1]. It was noticed that the twitter text was very unique in that the number of characters never exceeded 160. The news and blog entries had similar distributions, but blogs had much longer tails with certain entries over 20000 characters long.

Figure 1 Distributions of text lengths



It was suspected that the types of characters used were different in the different corpus samples. A further analysis was done on the distribution of non-alpha characters [Figure 2]. There does seem to be certain characters that may be indicative of the type of text being entered. The use of the exclamation point "!" was quite prominent in the twitter text.

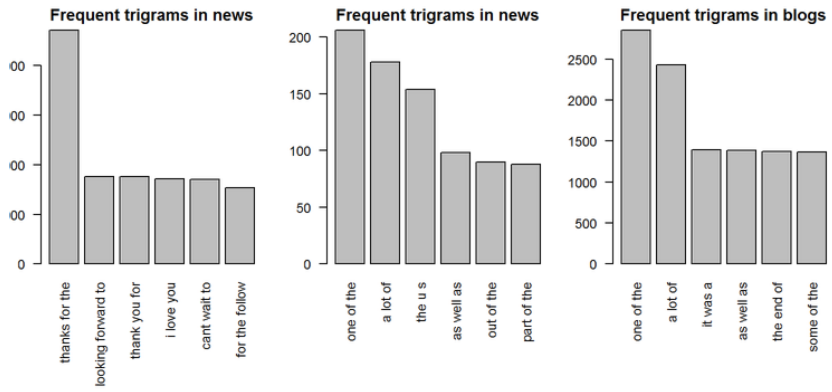
Figure 2 Frequent non-alpha characters



N-grams are frequently used in generating predictions [4] on test based data. As such, some general plots were created of the most frequently found trigrams (three word sequences) in the corpus [Figure 3]. It was noted that the twitter text in particular showed a marked difference in language to the other two document sources.

The data was stripped of all non-alpha characters and the frequencies of all trigrams in the sample corpus from the three files were generated.

Figure 3 Frequent trigrams

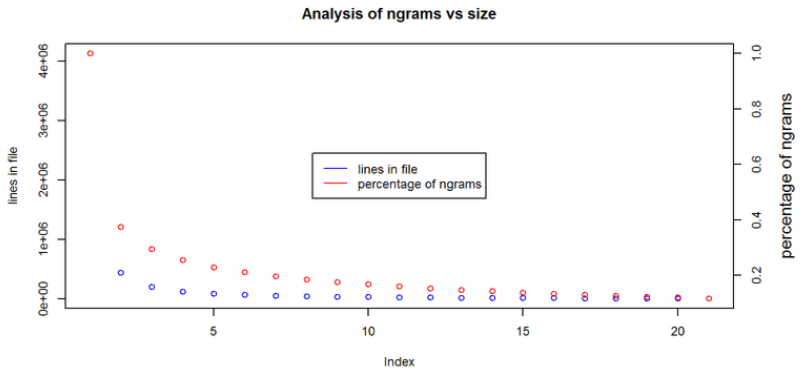


As the size of the files may be problematic in developing the final shiny app, I generated plots using the number of unique trigrams in the twitter sample document, and then pruned out the least frequent terms, starting with removing trigrams with 1 occurrence, then removing all trigrams with 2 occurrences, and continued pruning for occurrences less than 20. This information was plotted to get a sense of the approximate reduction in file size/ number of trigrams retained with various pruning levels [Figure 4].

A similar plot was generated with the total number of trigrams for the entire corpus as the corpus was trimmed to get an approximate idea of the loss of information with each pruning step, and joined to the plot below.

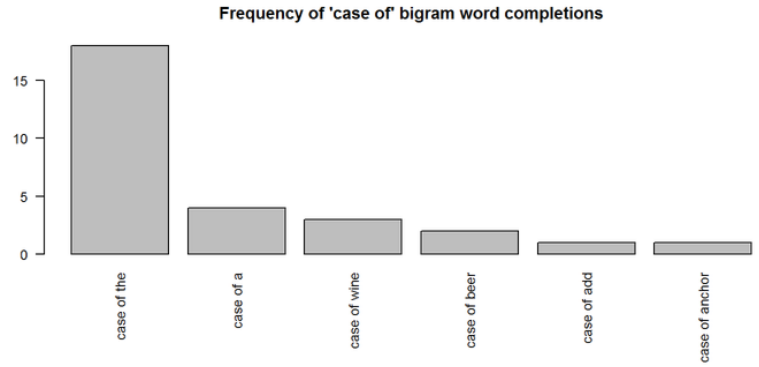
It was noticed that we can significantly reduce the size of the data file needed by dropping the least frequent occurrences of trigrams, but we will not be able to predict many infrequent occurrences of a given final word in a trigram.

Figure 4 File size and information available as infrequent trigrams removed



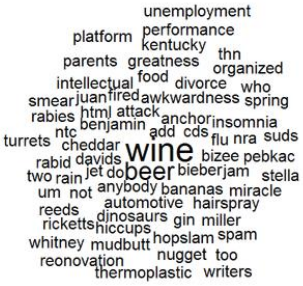
Using the first question in the Coursera Data Science capstone Quiz 2, it was decided to do an analysis of the phrase *"The guy in front of me just bought a pound of bacon, a bouquet, and a case of"* to see if trigrams had predictive capacity for this type of problem. The trigrams and three respective frequencies in the twitter corpus for the partial phrase *"case of"* were plotted to see which words might potentially complete the phrase [Figure 5].

Figure 5 Frequent trigrams beginning with *"case of"*



Additionally a word cloud [6] was created of all the trigrams for the partial phrase with common stop-words removed, as they are sometimes used to visualize text data, and it seemed a fitting completion to the exploratory analysis [Figure 6].

Figure 6 Most frequent words occurring after bigram *"case of"* in twitter sample



Conclusion:

My brief exploratory analysis shows that there may be some merit in trying to decide if incoming text is more twitter like or more news/blog like before attempting word prediction. My strategy at this point will be to predict whether the text is a tweet or news/blog text on some of the features explored here: length of text phrase, non-alpha character distribution and the largest frequencies of the words and n-grams used. Each bigram entered by the user will be looked up in either a twitter or news/blog data-frame and the most frequent words following the provided bigram will be returned as prediction. If no suitable match is found, the prediction will be based on the most frequent word associated with the unigram prior.