Capstone - NLP Assignment

Manuel Cerda

2/13/2018

Summary

In the last decade, the advances in machine learning has made it possible to model the Natural Language Processing (NLP) within smart devices so that we can have contextual information available in real time.

During the course of this Capstone Project, we will be creating an NLP model that will predict what we want to say based on the data that we feed in. That information comes from Blogs, News and Twitter, thanks to data distributed by SwiftKeys.

We will also include a Shiny App for the purpose of demonstrating the model.

Data Source

The training data for this project was downloaded from Coursera-Switfkeys.

It contains the following files: 1. "en_US/en_US.blogs.txt" 2. "en_US/en_US.news.txt" 3. "en_US/en_US.twitter.txt"

Data Processing

Exploratory Analysis

We need to explore the files before we try to load the Corpora that will be used in the model. Hence, we would like to know the file's meta data like the size of the files, the number of lines, the word count in total and the length of the line with most characters.

Because It would take a considerable amount of time and resources to process these files, and for the sake of our training data, we will subset more or less 1,000,000 of the characters from each file.

Selected Train Data

Filename	Size (in Bytes)	Number of Lines	Total Word Count	Largest Count (in Chars)
en_US.blogs.txt	1011455	4406	180235	1004003
en_US.news.txt	1007509	4932	168615	1004870
en_US.twitter.txt	1016462	14622	187417	1014581

Table 1. Description of the text files.

After selecting the training data, we will prepocess it to remove punctuation and numbers, fix white spacing and letter casing, as well as remove stop words.

N-Gram Modeling

The n-gram will be built with n=3 meaning that phrases will contain as much as 3 words, hence we will be able to visualize its frequency of ocurrence as shown in the following figures.

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed, please
```

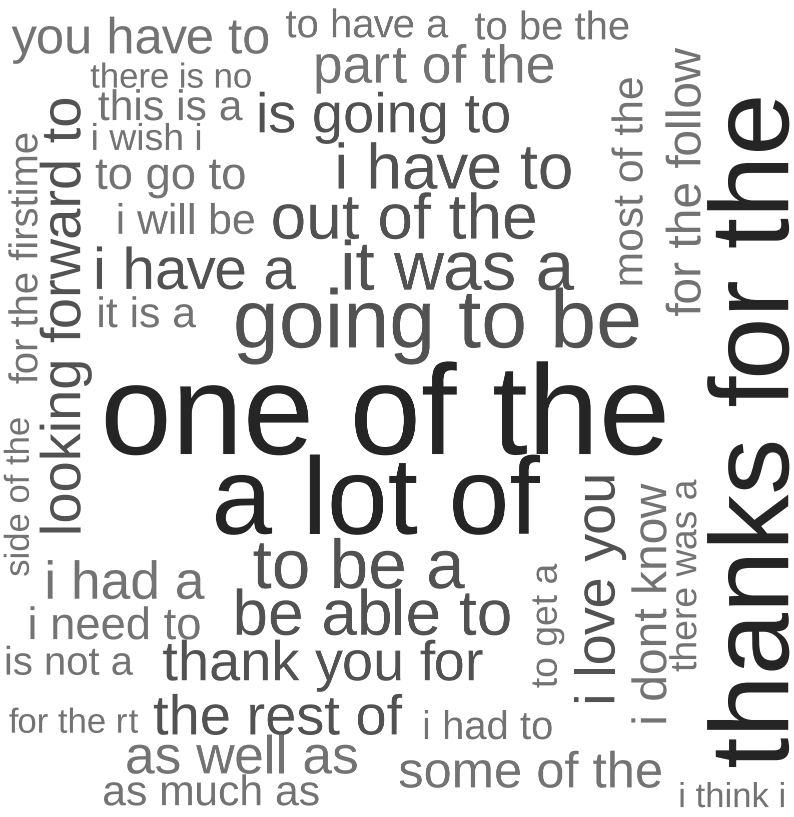

Figure 1. Wordcloud representation of the 3-grams model.

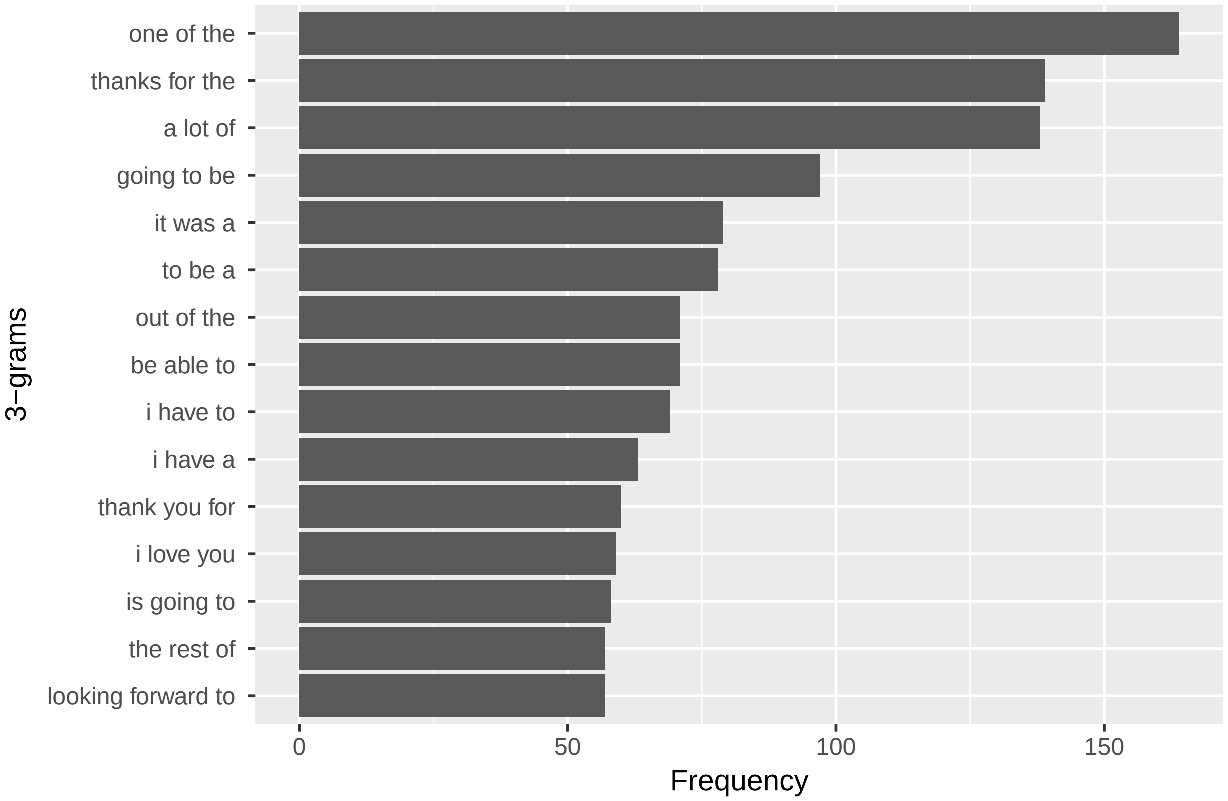

Figure 2. Frequency histogram of the 3-grams model.

Shiny App

The Shiny App will have a simple text field to gather the user input, It will also include a reacting event for observing what the user is typing so that It can feed the model for predicting the next word.

For predicting the next word, the model will compute the greatest probabilty of ocurrence from n-grams of size 2, 3 and 4. By doing the later, we will be using the Katz's Back Off algorithm for estimating the conditional probability of a word given its history in the n-gram.