Overview

Abinashi Prakash

19 July 2019

## Include Library

. ngram

. tm

. SnowballC

. Tidytext

. Rweka

. Wordcloud

. Quanteda

## Assignment Overview

As a Capstone project, we have developed the Shiny App which will Predict the next possible word based on the User input. This course will start with below high level steps.

1. basic analysis of large corpus of text documents to discover the structure in the data and how words are put together.
2. It will cover cleaning and analyzing text data.
3. building and sampling from a predictive text model.
4. build a predictive text product.

## Steps

1. Download Data in local folder.
2. Capture Sample data (2000) from three datasets.
3. Perform cleaning steps.
4. Tokenization.
5. Word Prediction Algorithm steps.

## Data Introduction

In this assignment,we are going to use “Corpus” called the “HC Corpora” dataset which is available at <https://d396qusza40orc.cloudfront.net/dsscapstone/dataset/Coursera-SwiftKey.zip> . The corpora are collected from publicly available sources by a web crawler. The crawler checks for language, so as to mainly get texts consisting of the desired language.We have datasets available in four languages such as , English,German,Russian and Finnish. In our analysis we will be using dataset available in English which is having three files as,

1. Blogs.
2. News.
3. Twitter.

Due to large dataset,it is advisable to perform exploratory analysis on Sample data.In this assignment we are going to consider sample size of 2000.

## A. Downloaded the data using encoding as UTF-8

Download data from <https://d396qusza40orc.cloudfront.net/dsscapstone/dataset/Coursera-SwiftKey.zip> to local working directory.

Download Profanity/ofensive list of Words from <https://www.freewebheaders.com/full-list-of-bad-words-banned-by-google/> .It has all the words banned by Google.

## B. Capture Sample data (2000) from three datasets.

Due to capacity limitation,we are taking the random sample of 2000 from each source files.

## C. Cleaning of Data using TM package.

### 1) Use Corpus

We will be using TM package to perform text mining using “Corpus” .We are going to use Volatile Corpus (Vcorpus) as once the R object is destroyed, the whole corpus will be gone.Also,with Vcorpus,predefind source format “VectorSource” is being used which only accepts (character) vectors.

### 2) Cleaning Data

This involves removing special characters, punctuations, numbers,whitespace,stemDocument,Offensive words and changing the text to lower case.We re not removing Stopwords for this assignment.

## D. Tokenization

Three\_gram : Group three cleaned words and save the sorted result based on number of occurences in Three\_gram.Rdata file. Two\_gram : Group two cleaned words and save the sorted result based on number of occurences in Two\_gram.Rdata file. One\_gram : Capture number of occurance of each word and Save the sorted result in One\_gram.Rdata file.

These files will be used in Word Prediction algorithm.

## E. Word Predictor Algorithm and guide

1. Put input word in Sidebar Panel.
2. Convert the input word in lower case,remove punctuaions along with numbers if any and split in separate columns.
3. If cleaned Input word/s having length greater than or equal to 2,the next predicted word would be value of most frequent 3rd column in Three\_gram file if below condition is true, . Last Input word should match with 2nd column of Three\_gram file and Second Last word should match with 1st column of Three\_gram file.
4. If condition in step:c is false then it will be searched in Two\_gram file and next predict word would be the most frequent 2nd column in Two\_gram file if below condition is true, . Last Input word should match with 1st column of Two\_gram file.
5. If cleaned Input word/s having length equal to 1,the next predicted word would be value of most frequent 2nd column in Two\_gram file if below condition is true, . Input word should match with 1st column of Two\_gram file.

\*\* In MainPanel,we are displaying 10 most frequent probbale words.

\*\* Source files Three\_gram,Two\_gram and One\_gram files are available in github repository