Next Word Prediction App

Introduction and Background

- Speed up text input and typing in constrained environments like a mobile phones or small keyboards
- Help people with writing difficulties
- Help people with inadequate language skills or while using a foreign language.
- Use as a supervised or unsupervised text generator
- Improve written communication quality by detecting mistakes and reducing grammatical and spelling errors.

Approach and Methodology

- The training and testing data was derived from sampling data from Twitter, Blogs and News. The data used is available at **https://d396qusza40orc.cloudfront.net/dsscapstone/dataset/Coursera-SwiftKey.zip**
- Due to the large volume of data, a 10% random sample of the three individual datasets was extracted and combined to train the model. A 1% sample of the combined dataset was used to test the model, quantify the accuracy and do cross-validation.
- Various optimization techniques were used to find the best balance between accuracy, performance and data volume.
- The prediction was done after reducing the data to 5-grams. An ngram is a set of contiguous words in a document. More details at **https://en.wikipedia.org/wiki/N-gram**

Approach and Methodology (Continued)

- The Backoff method was used to predict the probable next word. The user provided phrase is first searched for in the highest n-gram, if it is found present the next word, otherwise go the next lower level n-gram and search for the phase minus the first word, and so on till we reach the unigram. If it is not found in any n-gram then present the most frequently occurring single word. More details at **https://www.quora.com/What-is-backoff-in-NLP**
- The UI for the app was built using the Shiny package and it was deployed on shinyapp.io.
- A more detailed explanation of the methodolgy and suggestions on how to improve the accuracy and performance is available at **http://rpubs.com/joresh/368348**

Archiecture and Tools Used

- The code for the data loading, cleaning, transforming and reporting are all written in R.
- Generic functions have been developed for each step in the process which can be applied to any dataset (within contraints mentioned in the document at **http://rpubs.com/joresh/368348**)
- The R packges used include base R, dplyr, tm, tidytext, data.table and stringr.
- The app is deployed to the cloud using Shiny.

https://joresh.shinyapps.io/wordpred/