Untitled

MDC

November 13, 2015

Abstract

Introduction

Yelp is a businuess founded in 2004 that helps people to find local business; its mobbile application is available on the internet and on numerous mobil devices. The **Yelp Challenge Dataset** used in this project contains 1.5 million reviews of 61K business; the reviews from 366K users with 495K tips. The entire **Yelp Challenge Dataset** comprises 5 files – business, review, user, check-in, and tip; the file is in JSON format; details of the data elements are found at here.

The paper is organized as follows: (1) this seciton discusses the process of downloading data, reading JSON file formats into R objects; extracting only data for interested business at interested city; merging business and review data for business and review in focus of this paper. (2) section discuss how text data is being prepared for text mining; the proces includes remove punctuation, numbers, turn to lower case, steming, remove stop words. Generate *Document Term Matrix (dtm)*; word cloud is a useful tool to have a summary view of text of corpus of interest.

The source code is found here in github.

Preparing Data

Down load the dataset at the above URL. Unzip the file. *getJsonData.R* is the function that read a JSON file and convert it into a data frame. There are 5 data frames: business (*businessDf*), review (*reviewDf*), user (*userDf*), check-in (*checkinDf*), and tip (*tipDf*). Since each of the dataframe is very large (larger than than a class project data), the data is saved into the file system, which is stored in the directory *./processData/* in the github. This is done to reduce that time parsing; for each use, one only need to load() the process data into the R-global environment to use.

In order to make a quick and fast initial exploratory analysis, *businessDf* is persisted into a SQL Server database. The following tables show the distribution of number of business categories contains terms like '%physician%', '%doctor%', '%medic%', '%health%, and not'%market%'('%' is SQL wildcard syntax). The following table shows the distribution of number of businesses for each state that satisfy the above conditions (see function *getDataFromDb.R* in getAndPrepDAta.

```
##    state number_of_reviews
## 1     RP                1
## 2    MLN                1
## 3     SC                2
## 4     QC               12
## 5     IL               17
## 6    EDH               33
## 7     BW               34
## 8     WI               65
## 9     PA               67
## 10    NC              149
## 11    NV             1065
## 12    AZ             1786
```

From the below table, **reviews for Medical related businesses in the states AZ, NV, and NC are included in this analysis.** *AZ_df, NV_df, NC_df* are the dataframes that contain the merged reviews and business information for the healthcare related business in the three states Arizona, Nevada, and North Carolina. R code that generates these the dataframes is found in *combineBusinessReview.R*, in the same getAndPrepDAta. *For state AZ, there are 12,358 review text; 9,006 reviews for NV state, and 940 reviews for NC state.*

Preparing data for text mining

In order to prepare text for mining purpose,a *corpus* is generated from the text. After which, punctuations are removed, numbers are removed, text is converted to lower cases. In addition, text will be stemming, and stop words will be removed. For this exercise, English stopwords will be using. Refer to function clean(), for details of this process. R *tm* package provides all functions to do the cleaning process.

When the corpus is 'cleaned', the using *tm* package to generate document term matrix (dtm) or term document matrix (tdm). DTM is a matrix of m-term of represent the vocabulary of the text data set as columns, and n-document rows; where as the tdm has n-document columns, and m-term rows. Refer to *prepCorpus_dtm()* and *prepCorpus_tdm()* here for codes.

Sparsity All dtm's and tdm's are quite sparse. After number of trials in removing the sparsity, the dtm's and tdm's achieve 78-79 percent sparsity, and the vocabulary contains from 47-53 terms.

Wordcloud for AZ review text is shown below, on the left, the text in DTM is sparse, on the right some sparsity is removed, about 78% sparsity.

Following wordcloud shows the very sparse reviews DTM for the state of AZ, and its wordcloud when DTM sparsity is removed to 78%.

```
##          [,1]
## i        Integer,603112
## j        Integer,603112
## v        Numeric,603112
## nrow     12358
## ncol     20796
## dimnames List,2
## attr(,"class")
## [1] "DocumentTermMatrix"    "simple_triplet_matrix"
## attr(,"weighting")
## [1] "term frequency" "tf"
```

Methods

Discussion

References