Text mining using LDA and LSA

My D. Coyne

November 13, 2015

Introduction

Yelp is a businuess founded in 2004 that helps people to find local business; its application is available on the internet and is accessble via numerous mobile devices. This makes an ideal application for visitors to read/write reviews, find events, or talk to other "Yelpers". It is very exciting that Yelp makes available the **Yelp Challenge Dataset** to the Data Science Specialization Capstone project for data analysis purpose. The Yelp Challenge Dataset comprises of 5 files – business, review, user, check-in, and tip, where there are 1.5 million reviews of 61K business; the reviews from 366K users with 495K tips. Details of the data elements are found at here.

Using the *business and review datasets*, I aim to answer the following questions for my text analysis. It seems to be natural that a reviewer gives a high score then it is an indication of a good service rendered at a business; similarly, a low review score accompanied with uncomplimentary service received by the reviewers. **Would it is possible to qualitatively determine the review score using text analytics? In other words are there particular terms indicative of the score? The application of the analysis may be used to help Yelpers to retrieve better results from searching reviews. The analysis focuses on medical doctor, physician, dentist or heatth centers services.**

The paper is organized as follows: (1) seciton discusses the process of obtaining, cleaning, and preparing data analysis. The (2) section discusses findings. Finally, the (3) section will interpret the reuslts and answer the above questions.

The source code is found here in github.

Methhods and Data

Preparing Data

Downloaded the datasets at the above URL, after unzipped, *getJsonData.R* is the function that read a JSON file "flatten" it into data frames. There are 5 data frames: business (*businessDf*), review (*reviewDf*), user (*userDf*), check-in (*checkinDf*), and tip (*tipDf*). In this project, since "raw" and process data are not only large, but also takes long computing time to produce; the intermidiate data R objects are persisted into file system and loaded into R global environment as needed. Database tables are also utilized for peristence and queries as well.

The following table shows the distribution of number of businesses, grouped by state, that offers medical doctor, healthcare center, dentists, chiropractor services (see function *getDataFromDb.R* in getAndPrepDAta github.

```
##    state number of health related business
## 1     RP                                 1
## 2    MLN                                 1
## 3     SC                                 1
## 4     QC                                 7
## 5     IL                                15
## 6    EDH                                30
## 7     BW                                 33
## 8     WI                                 53
## 9     PA                                 60
## 10    NC                                128
## 11    NV                                987
## 12    AZ                               1614
```

From the above table, **reviews for Medical related businesses in the state Arizona (AZ) are included in this analysis.** An ambitious intent in comparing medical review services from Arizona, to Nevada (NV) to North Carolina (NC) was planed, but due to the space limitation, analysis for NV and NC states was removed. *AZ_df* is the dataframes that contain the merged reviews and business data items for the healthcare related business. R code that generates these the dataframes is found in *combineBusinessReview.R*, in the same getAndPrepDAta. *For state AZ, there are 1,614 medical and healthcare providers, with 10,827 review text to form the corpus.*

Preparing data for text mining

In order to prepare text for mining purpose, all review text for AZ generates a *corpus*. After which, text is convert to lower case with punctuations, numbers and English stopwords are removed. Stopwords are words like 'a', 'the', 'ive','we', etc. The words in the corpus are also stemed; for instance, 'am', 'are', 'is' has a stem 'be'. An other example is the text may contain "The boy's cars are different color"; after stemming process, the text may be converted to "the boy car be differ color" Refer to function clean(), for details of this process. R *tm* package provides all functions to prepare text for the mining process that follows.

After 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 x n-document); whereas the TDM is a transpose matrix of DTM, or (n-documents x m-terms). Refer to *prepCorpus_dtm()* and *prepCorpus_tdm()* for R code, here in github

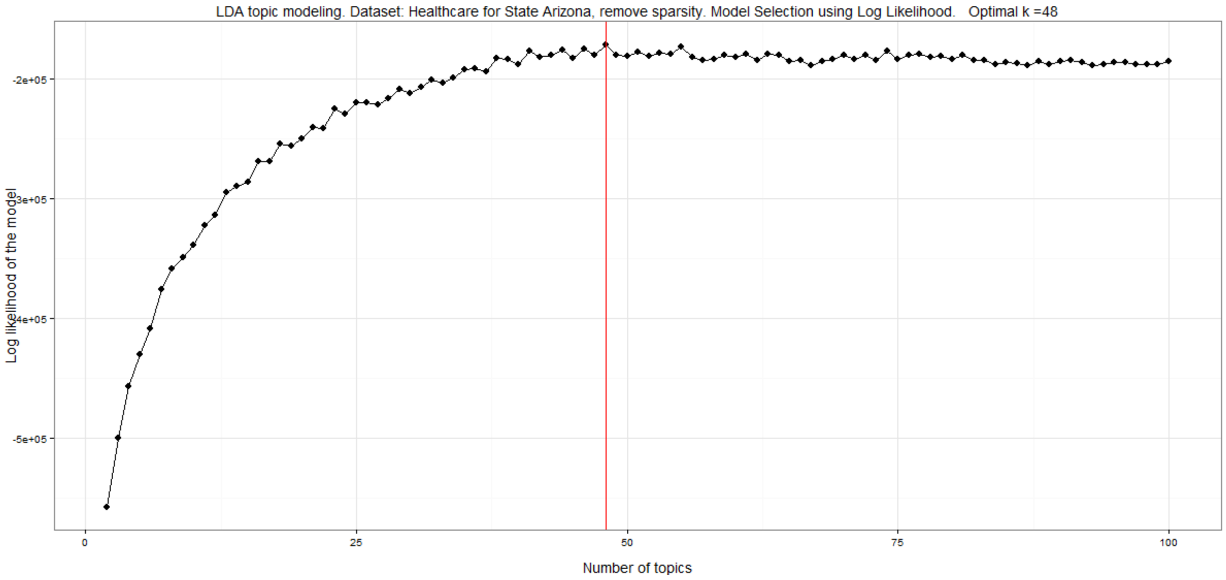
Sparsity Both dtm's and tdm's for this text corpus is 100% sparse. After number of trials in removing the sparsity, the dtm's (hence tdm's) achieves 78-79 percent sparsity. At this level of sparsity, the vocabulary contains from 47-53 terms. **In this analysis, 78% sparse DTM, with 47 terms for its vocabulary is used for further text mining purpose.**

Methods

Latent Dirichlet Allocation (LDA) Topic Modeling

In this analysis, LDA is used to classify all reviews for healthcare and medical business in AZ with a topic. **Topic Models** are "[probabilistic] latent variable models of documents that exploit the correlation among the words and laten semantic themes" (1). Topic ("latent") is hidden; it is to be estimated. Topics link words in a vocabulary and their occurrence in a document. A document is seen to be a mixture of *topics*. LDA relies on the *bag-of-words* assumption, which means that that words in a document are exchangeable and their order, therefore, is not imporant. LDA has been thouroughly explained; in this project, I will use R implementation of LDA to perform analyses and will not go into the mathematical foundation of LDA (nor LSA in the next section). In this project, *LDA is used to generate the topics for all review text of all medical and healthcare business of AZ state as found in the Yelp Challenge Dataset.* The aim is to use LDA to assist with sorting out the reviews in an automatic and fast way.

For LDA model selection, the maxinum log likelihood occurs at the **optimum number of topics of 48**, as shown in the following figure.



Latent Semantic Analysis (LSA)

The Laten Semantic Analysis (LSA) model is a theory for how meaning representation might bee learned from encountering a large samples of language without explicit directions as to how it is constructed. LSA assumes that the meanin of sentences is assumed to be the sum of the meaning of all the words occuring. Hence the meaning of multi-word phrases is more greatly determined by which words occur in the phrase, rather than how the words are ordered. A second assumption is that the semantic asssociatons between words is *latent* in a large sample of language and *eventually* the meaning is learned. LSA is used in this project to find the **similarity** of review text.

Results

LDA finds Latent Topics for Review Text

All 10,827 review text of 1,614 medical and healthcare business establishments in Arizona state are sorted out into **38** topics; which is the optinum number of topics, using maxinum log likelihood. Following table shows the topic number and the five highest score terms that make up the topic.

```
##    lda_topic                           lda_terms
## 1          1   best,staff,friend,recommend,ive
## 2          2         even,never,first,just,one
## 3          3           ive,year,never,now,best
## 4          4      back,come,get,just,recommend
## 5          5            get,will,went,just,one
## 6          6          see,doctor,offic,need,just
## 7          7           dont,know,even,take,now
## 8          8           call,back,one,well,feel
## 9          9 friend,staff,recommend,good,great
## 10        10          just,like,dont,know,look
## 11        11           one,even,went,just,know
## 12        12       appoint,even,day,also,visit
## 13        13       wait,time,take,back,patient
## 14        14          back,went,time,will,come
## 15        15        realli,like,feel,place,one
## 16        16      patient,care,experi,need,get
## 17        17    great,recommend,best,feel,help
## 18        18     offic,can,appoint,like,realli
## 19        19            make,feel,also,day,get
## 20        20            need,get,just,work,see
## 21        21        will,come,know,dont,experi
## 22        22      care,staff,patient,alway,now
## 23        23              one,get,day,can,went
## 24        24           get,just,need,time,know
## 25        25          look,like,know,just,good
## 26        26 staff,friend,great,recommend,feel
## 27        27        day,call,went,need,appoint
## 28        28        well,good,realli,also,come
## 29        29        good,experi,just,look,work
## 30        30           can,make,get,will,first
## 31        31 staff,friend,recommend,experi,get
## 32        32       place,realli,make,dont,need
## 33        33      work,realli,need,never,visit
## 34        34 experi,great,recommend,friend,now
## 35        35          time,take,also,first,now
## 36        36          care,take,need,well,dont
## 37        37           like,feel,just,good,now
## 38        38            see,now,can,make,first
## 39        39      doctor,patient,day,went,good
## 40        40         visit,first,time,one,need
## 41        41          alway,year,best,ive,will
## 42        42    offic,staff,friend,feel,experi
## 43        43           never,will,can,went,ive
## 44        44      time,first,come,place,realli
## 45        45        also,work,help,realli,good
## 46        46    help,recommend,also,staff,need
## 47        47           year,now,well,staff,see
## 48        48        time,wait,appoint,dont,one
```

Comparing reviews for business

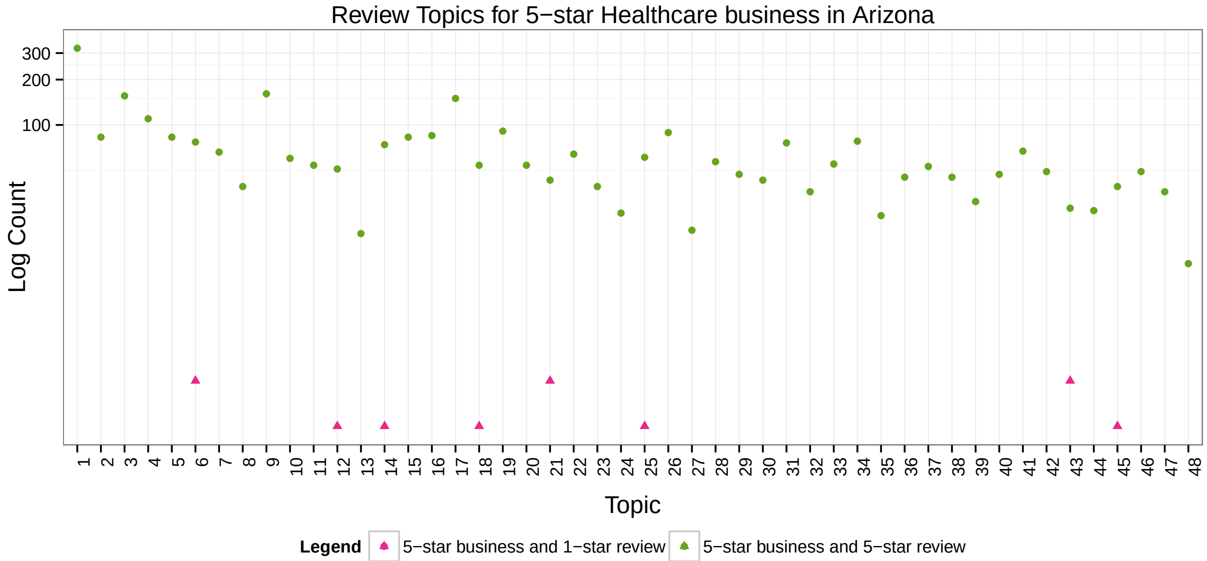
Contrasting 5-star reviews and 1-star reviews for 5 and 1 star business establisments with the two graphs shown below.

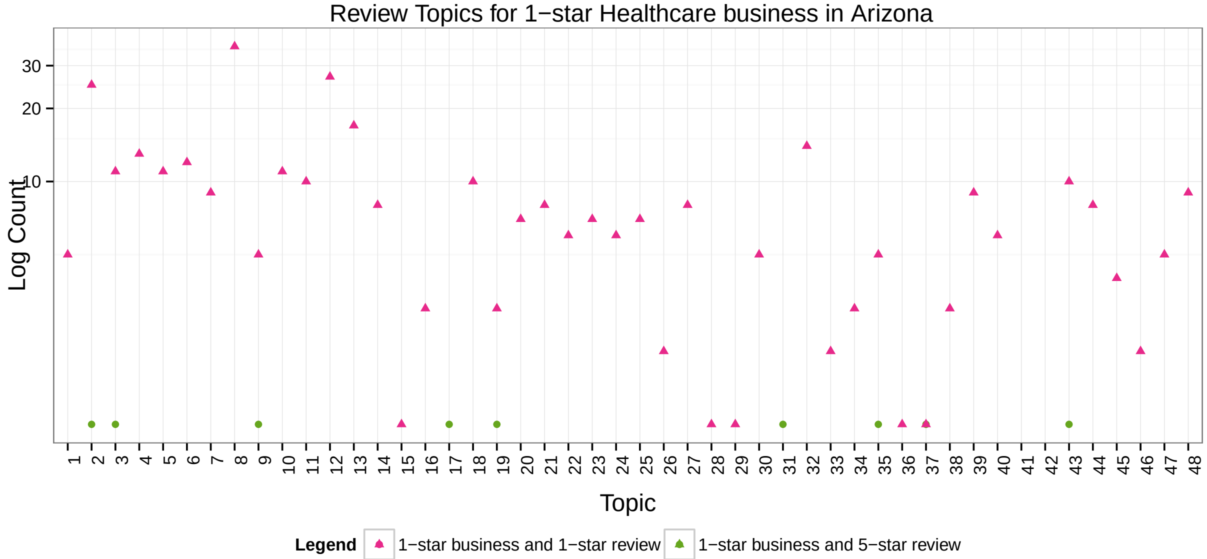
a. *Topic 1 with terms "best, staff, friend, recommend"*

b. *Topic 9 with terms "friend, staff, recommend, good, great"*

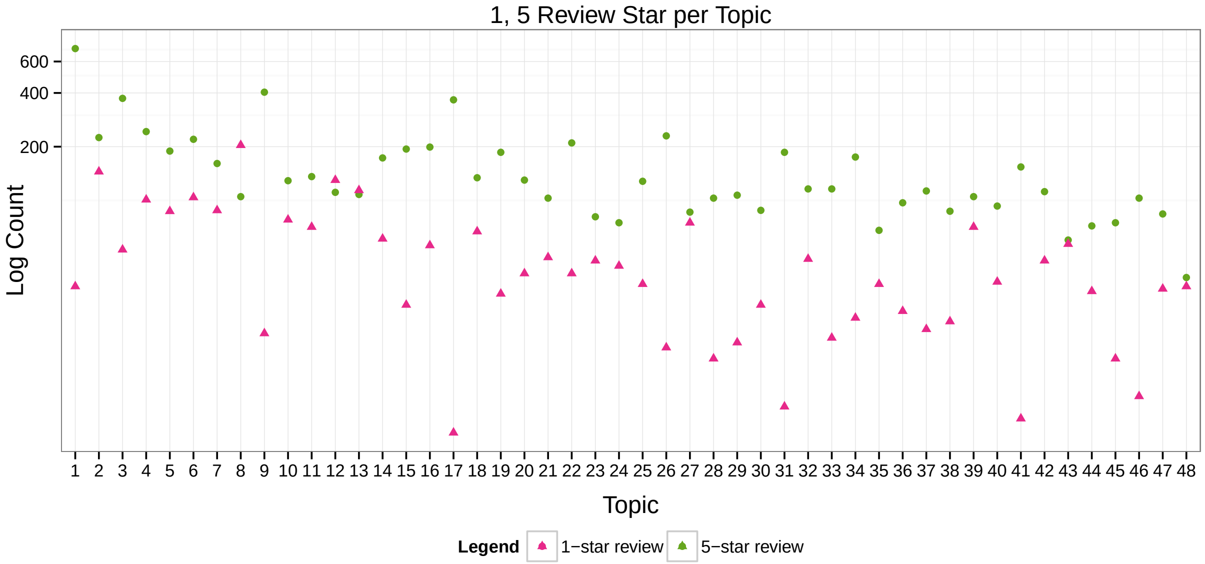
c. *Topic 17 with terms "great, recommend, best, feel, help"*

are strong indicators of great reviews.





By removing the business ranking from the above two graphs, the following graph shows 5-star vs 1-star reviews for *all ranking/star* businesses. The graph **confirms that Topic 1, topic 9, and topic 17 are indicative of best ranking reviews.**

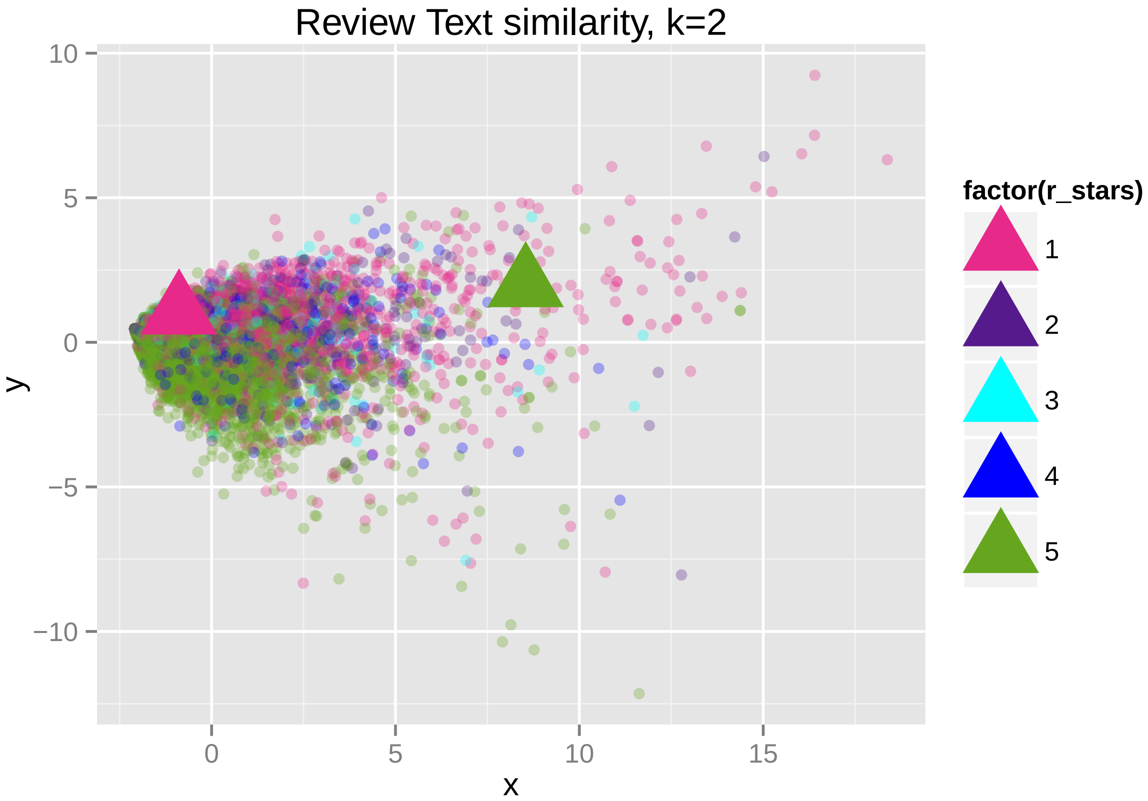


In addition, by oberving the above graph, *Topic 8 with terms "call, back, one, well, feel"* shows high counts for 5-star reviews, as well as for 1-star reviews, with low star reviews are higher than high stars. Below table show two specific cases where both reviews were placed in the same topic #8, but the reviews are completely different; yet the topic with terms 'call', 'back' fits well. Latent Semantic Analysis (LSA) is used to review the document similarities.

Business ID	Review ID	R star	Example text
1eCvpgvB4QA-0fSwb8-5Dw	NbcYFZRNBAlkzJHWtKgpZQ	5	Within 1 hour, he called me back
e4FM01_iF_2LLN_yiaYcHA	-d1Sl2KzWUIBsXOxH_0jdQ	1	heard nothing. . . called twice. . . nothing

LSA Document Similarity

Due to space limitation, only the result of LSA is shown below. LSA space for all review text, distance matrix are calculate. LSA space dimension is reduced to 47 terms. Below figure shows an example of how the review text are 'clustered', or similarity. **The above two text reviews, although placed in the same topic by LDA, are very far apart by LSA calculation.** The two review documents are marked by two emphasized triangles in the figure below.



Discussion

1. It is qualitatively possible to sort out the review text with appropriate topic assigned. For the Yelp Challenge Dataset, the terms *"best,friend, recommend, good, great, help"* are strongly associated with great reviews. However LDA fails to detect negation, as seen in the cases when two different reviews are placed in the same topic 8, with terms "call, back, one, well, feel". A case when the Yelper gave 5-star review dues to the business 'call back promptly as promised"; whereas another case, when the Yelper"did not get the call back, hear nothing", hence the business received a 1-star review.

2. LSA helps in finding similarities or dissimilarities in document text. The review text is used to form a corpus, from which document term matrix (dtm) was built with 78% sparsity. At this sparsity, the vocabulary for the corpus conprises of 47 terms of the LSA space. Within the LSA space, it uncovers that the two text reviews are quite far apart in LSA space; they are not 'close' enough in distance to be in the sme cluser, or in the same topic as resulted by LDA.

3. LSA can be used for reducing dimensionality and served as a tool to perform similarities between documents, hence clustering.

4. Although LDA and LSA can compliment each other in text mining, it requires a considerable human effort in reviewing the results; they do not provide a turn-key solution.

5. This work can be further improved in applications such as:

   a. Intelligently Categorize business type, categories that are currently in Yelp using LDA

b. Forming a search engine where users can use natural language and better information retrieval. In this applicaiton, review text are grouped into its topics; after carefully review of the topics and its grouping, the text can be served as examples to train LSA in forming indices for thes topics (or categories). Then Yelpers can query these indices using natural language (as opposed to keyword search) to find business needed. The query is compared to the trained examples, the most *similar* reviews in latent or hidden meaning will be retrieved and present to users.

References

(1) D. M. Blei and J. D. Lfferty. A correlated topic model of Science. Annals of Applied Statistics, 1(1):17-35, 2007

(2) R. Arun, V. Suresh, C.E. Veni Mdhavan, and M.Narashima Murty. *On Finding the Natural Number of Topics with Laten Dirichelet Allocation: Some Observations, PAKDD 2010, Part I, LNAI 6118, 391-402.