Text Mining using LDA and LSA

My D. Coyne

November 13, 2015

Introduction

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.

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 business that provides medical services or health centers.** I aim to answer the above questions through text analysis techniques applied on *Yelp business and review datasets.*

The source code and this report are also on Github, https://github.com/mdcRed/Capstone. Details of the data elements in the dataset are found at http://www.yelp.com/dataset_challenge.

Methods and Data

Preparing Data

The Yelp data is in JSON format, use the function *getJsonData.R* to read a JSON file and "flatten" it into data frames. There are 5 data frames, this paper only focuses on the business (*businessDf*) and review (*reviewDf*). In this project, since the "raw" and process data are not only large, but also takes long computing time to produce; the intermediate data R objects are persisted into file system and loaded into R global environment as needed. Database tables are also utilized for persistence and queries as well.

The following table shows top three states that have the largest number of medical doctors, healthcare centers, dentists, and chiropractor business registered in Yelp. For the scope of this paper, review text for Medical related businesses in the **state Arizona (AZ)** will be analyzed.

```
##   state number of health related business
## 1    AZ                              1614
## 2    NV                               987
## 3    NC                               128
```

AZ_df is the data frames that contain the merged reviews and business data items for the healthcare related business. R code that generates these the data frames is found in *combineBusinessReview.R. For state of Arizona, there are 1,614 medical and healthcare providers, with 10,827 review text to form the corpus.*

Preparing data for text mining

All 10,827 review text documents are used to generate a *corpus*. After which, text is converted to lower case; removed of punctuations, numbers, and English stop words. Stop words are words such as 'a', 'the', 'ive','we',

etc. Then, the words in the corpus are stemmed. Stemming is a process to reduce all words in the corpus to its stem. For example, the text may read "The boy's cars are different color", becomes "the boy car be differ color" after the stemming proces. Refer to function *clean.R* for details of this process. The R *tm* package provides all functions to prepare text for the mining process that follows.

After the corpus is 'cleaned', the *tm* package is used to generate document term matrix (DTM) or term document matrix (TDM). DTM is a matrix of (m-terms by n-documents); whereas the TDM is a transposed of DTM, or (n-documents by m-terms). Refer to *prepCorpus_dtm()* and *prepCorpus_tdm()* for the R code.

Both dtm's and tdm's for this text corpus is 100% sparse. This is because the corpus has 10827 documents with 19,402 terms; and majority of the terms appear only once in all documents; hence increase the sparsity of the measure (or term in text mining). After number of trials in removing the sparsity, the dtm's (hence tdm's) sparsity is reduced to 78%-79%. At this level of sparsity, the terms are from 47-53 terms, respectively.

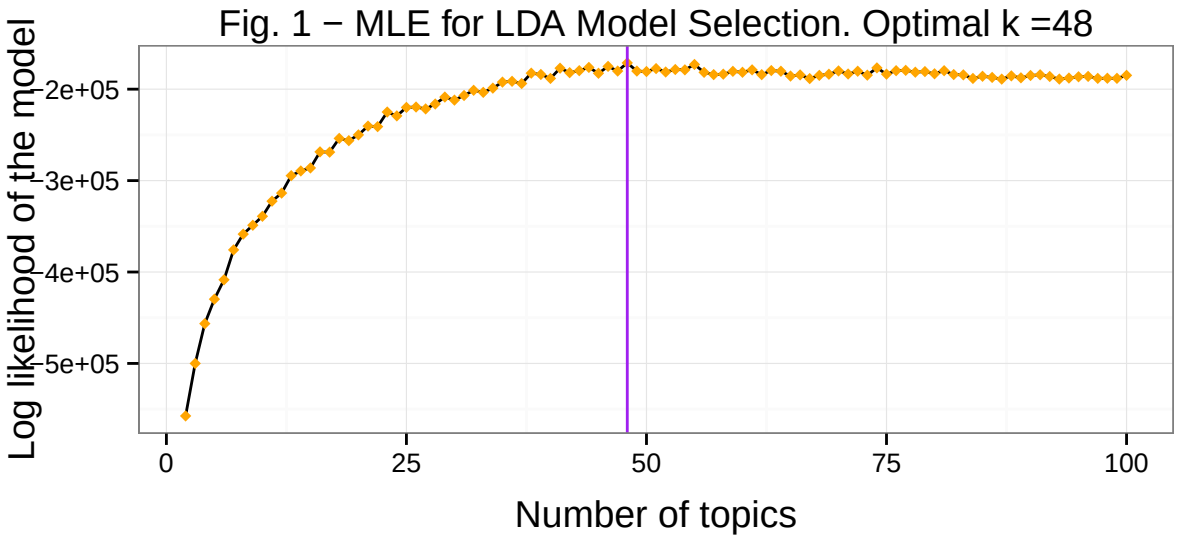
Methods

Latent Dirichlet Allocation (LDA) Topic Modeling

Topic Models are "[probabilistic] latent variable models of documents that exploit the correlation among the words and latent 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–words in a document are exchangeable and their order, therefore, is not important.

In this analysis, *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.*. Topic comprises of terms. After all reviews were labeled by a topic (set of terms), by cross reference the reviews to the number of stars associated with the review and the business, one may find the association of terms to the review numbers. This is a qualitative way to explain the review number.

For LDA model selection, the maximum log likelihood estimate (MLE) occurs at the **optimum number of topics of 48**, as shown in Fig. 1 below.



Latent Semantic Analysis (LSA)

The Latent Semantic Analysis (LSA) model is based on a theory for how meaning of text might be learned from encountering a large samples of language without explicit directions as to how the sentences are constructed. LSA assumes that the meaning of sentences is assumed to be the sum of the meaning of all the words occurring. 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 associations 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 documents.

Results

LDA finds Latent Topics for Review Text

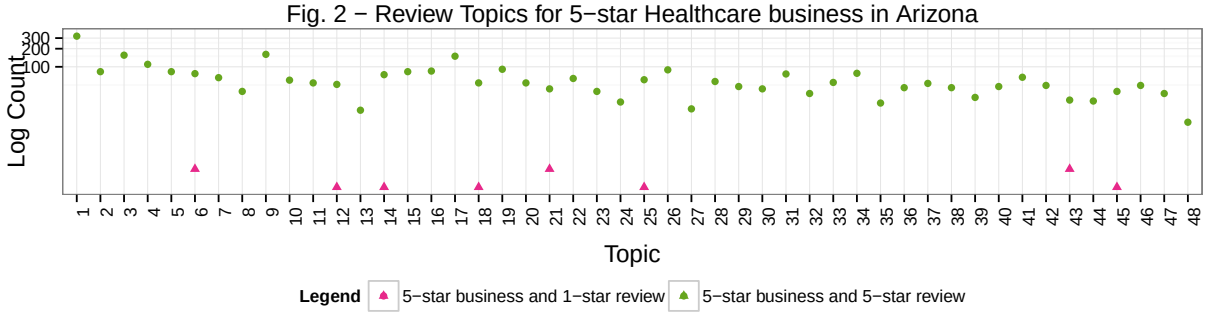
All 10,827 review text of 1,614 medical and healthcare business establishments in Arizona State are grouped into **38** topics. Following table shows the topic number and the five highest score terms that make up each topic; due to the space constraint, only selected topics that are used in the discussion are shown.

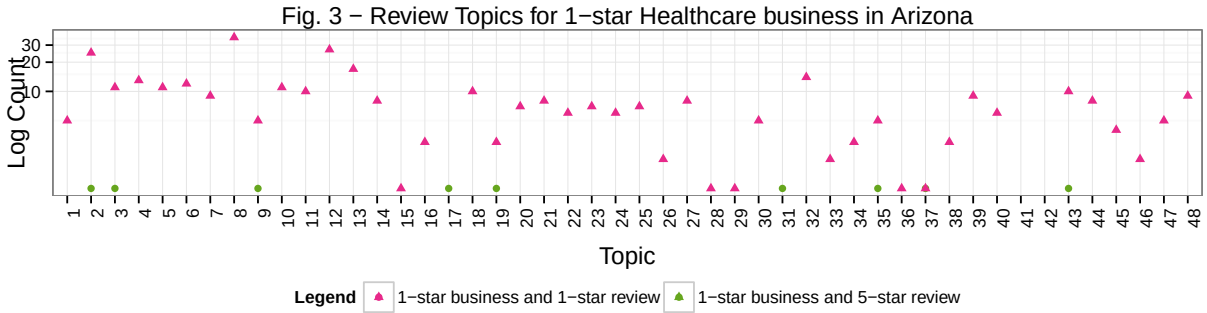
```
##   lda_topic                          lda_terms
## 1         1   best,staff,friend,recommend,ive
## 2         8           call,back,one,well,feel
## 3         9 friend,staff,recommend,good,great
## 4        17    great,recommend,best,feel,help
## 5        26 staff,friend,great,recommend,feel
## 6        31 staff,friend,recommend,experi,get
## 7        34 experi,great,recommend,friend,now
```

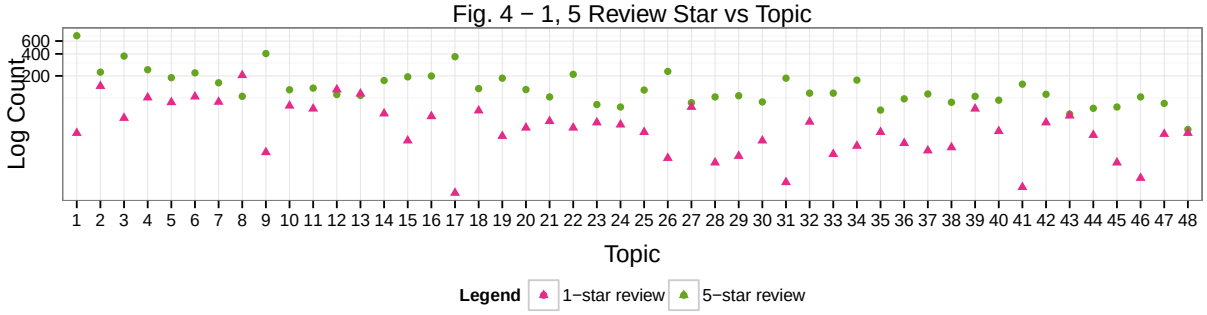
Comparing reviews for business

For contrasting purpose, Fig. 2 shows the 5 and 1 star reviews for 5-star business; Fig. 3 shows 5 and 1 star review for 1-star business. It is noted that topic #1, #9, #17, #31 are high in 5-star review, but low in 1-star review. This follows terms *"best, staff, friend, recommend, good, great"* are strong indication of good reviews.

By removing the business ranking from the above two graphs, Fig.4 shows 5 and 1 star reviews for *all* businesses. The wide range in log count on the graph **confirms that Topic 1, topic 9, and topic 17 are indicative of best ranking reviews.**







In addition, Fig. 4 also shows an interesting observation: some 1-star reviews are labeled with *Topic 8 with terms "call, back, one, well, feel", as well as 5-star review*. Furthermore, higher count of 1-star reviews is labeled with topic #8. This is to say that some terms are used indicating both desirable and undesirable services. By just reviewing the terms, it is not evident of the reason; one must look into the text. 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, in expect to complement to the LDA technique.

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

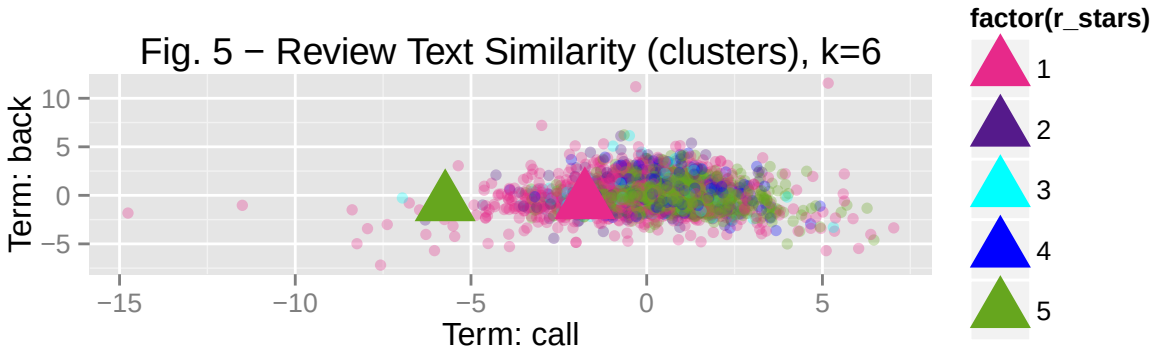
LSA Document Similarity

Due to space limitation, only the result of LSA is shown below. In practice the DTM is a m by n (m x n) matrix, where m is number of documents, and n is number of terms. DTM can be decomposed, by a singular Value Decomposition (SVD), into matrices M, T, and D. Then only k-eigen values are kept, the *approximation of the DTM, or LSA Space is represented below, with T is Term Vector matrix, D is Document Vector matrix and S is the single values.* For our specific case, *k is found to be 47 terms.*

```
LSA Space = Approximated of (M) =  T (m x k) . S (k x k) . D(k x n)
```

Each dot on Fig. 5 is a review text. The x and y axis is a dimension of LSA space, in this case is term 'call', and 'back', respectively. Fig. 5 shows the review text are 'clustered', or similarity. An observation is **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. The

1-star review (in pink) is within the cluster for 1-star riew; however the 5-star review (in green) is an outlier in its cluster.



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 both labeled as topic 8 ("call, back, one, well, feel"). Although the labeling is correct, a reviewer gives 5-star because the doctor 'call back promptly as promised"; whereas in the second case, when a user"did not get the call back, hear nothing" from the office, he gave a poor review of 1-star.

2. At the 78% sparsity of the corpus, R LSA package calculation reduces the dimensions to *47 terms*, or *47 eigen values*. using LSA, it uncovers that the two text reviews are quite far apart (shown as large triagles in Fig. 5). They are not 'close' enough in distance to be in the same cluster. This observation shows that the two pieces of text are made up of similar vocabularies, are labeled by same LDA topic, but quite 'far apart' in LSA space. *LDA and LSA can be used together providing complementary analysis tool to each other.*

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

4. 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 application, 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 these 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.