MCIT
Modeling, control & information technologies

Analysis of Correspondence Reviews Texts and Stars Ratings

https://doi.org/10.31713/MCIT.2020.27

Tetyana Sopronyuk
Yuriy Fedkovych Chernivtsi National University
Chernivtsi, Ukraine
t.sopronyuk@chnu.edu.ua

***Abstract*—The article analyses the connection between the clients' reviews of the texts and their published ratings [1]. To realize the task the language R and the packages such as jsonlite, tm, SnowballC, wordcloud, RColorBrewer, ggplot2 and others have been chosen. We used the data of the service Yelp, opened for academic goals and analysis [6]. For the analysis we used sentiment analysis. Sentiment analysis is the computational treatment of opinions, sentiments and subjectivity of text [2].**

Keywords—**Sentiment analysis, Emotion detection, Yelp, corpus, textual review, positive words, negative words, clouds of the words**

I. Introduction

Every day many visitors generated their reviews and determine the sentiments of these contents can be useful for business companies and individuals.

The matter deals with the description of the question/problem and the rationale for studying it: we tried to find out, if it was possible to analyze the sentiments of the reviewers [1-3] to define the kind of rating in a certain type of business. Can we predict the sentiment of a textual review (positive or negative words) from a corpus of restaurant and bar businesses reviews?

Using a subset of the data for the last 5 years and random samples of reviews we try to predict the number of stars from 1 to 5 from the review text.

Yelp, an American multinational corporation, is located in San Francisco, California. It deals with the development of hosts and markets Yelp.com and Yelp mobile app, publishing the collections of peoples' reviews about local businesses, reservation SeatMe and food delivery Eat24 services online. Yelp is providing all the data and reviews of the 250 closest businesses for 30 universities for students and academics for research [6].

The company trains small businesses to react to reviews, arrange social events for reviewers, it also supplies data businesses, and health inspection.

II. Methods and Data

A. Input Data for Sentiment Analysis

The data in the Yelp academic dataset are stored in a single json files. We used two files from the dataset which were provided by Yelp (*yelp_academic_dataset_review.json*, *yelp_academic_dataset_business.json*). Further we connect these tables by the key and bind them by the key of business_id.

The key corresponds to the business, dealing with bars and restaurants. Such business is widely used in the Yelp service. Since the sets of data are very big in volume, we restricted ourselves only by the part of the data. (We consider the notes, chosen for the last 5 years).

Then select the subset from this data and get 30% random records from this data. For different sets of random data we spend sentiment analysis. Analyzing the frequency of usage the positive and negative words, we find the means, median and the total number of the most commonly used words. This process repeats in each of 1-5 star reviews.

The code in the language R for the loading data is given below for the analysis and narrowing scope.

```
# loading review data
review_data <-
stream_in(file("yelp_academic_dataset_review.json"))

# loading business data
business_data <-
stream_in(file("yelp_academic_dataset_business.json"))

# selecting columns stars, text, business_id for last 5
year reviews
review <- subset(review_data, review_data$date>"2012-01-
01")
subReview<-data.frame(review$stars, review$text,
review$business_id)
names(subReview)<-c("stars","text","business_id")

# selecting the Bars|Restaurants categories
business_id <-
business_data$business_id[grep("Bars|Restaurants",

business_data$categories)]
subBusiness <- data.frame(business_id)
```

```
# merging subReview and subBusiness by="business_id"
Reviews <- merge(subReview,subBusiness, by="business_id")

set.seed(848)
# random subsampling without replacement (30%)
subsamples=
sample(1:nrow(Reviews),size=nrow(Reviews)*0.3,replace=F)
partReviews <- Reviews[subsamples, ]
```

B. Calculation

Further the program implements the following actions:

- files with positive and negative words are loaded;
- reviews, depending on the parameter, sent into function, are selected (the number of stars);
- the punctuation, articles, prepositions and rarely used words are neglected;
- the corpuses of words are formed from the remaining words;
- the sets of positive and negative words are created from the words corpuses, found in the reviews;
- the main statistic characteristics (means, medians and sums of the proportion positive-negative) are calculated.

The next fragment of the code realizes sentiment analysis, described above.

```
pos=scan('positive-words.txt ',what='character',comment.ch
ar=';')
neg=scan('negative-words.txt',what='character',comment.ch
ar=';')

library("tm")
proportion<-function(countStars)
{
    reviewStars<- partReviews[ partReviews$stars==countSt
ars,]$text

    # converting the relevant part of your file into a co
rpus
    myCorpus<-Corpus(VectorSource(reviewStars))
    myCorpus <- tm_map(myCorpus, content_transformer(tolo
wer))
    myCorpus <- tm_map(myCorpus, PlainTextDocument)

    # removal punctuation
    myCorpus <- tm_map(myCorpus, removePunctuation)

    # removal common words like "a", "the" etc
    myCorpus <- tm_map(myCorpus, removeWords, stopwords("
english"))
    myCorpus <- tm_map(myCorpus, stemDocument)

    # turning the corpus into a document term matrix
    dtm <- DocumentTermMatrix(myCorpus)

    # extracting frequently occuring words
    notSparse <- removeSparseTerms(dtm, 0.99)

    # most frequent words remain in a dataframe, with one
 column per word
    finalWords<-as.data.frame(as.matrix(notSparse))

    # extracting positive/negative words from all final w
ords
    extractPos<<-intersect(names(finalWords) , pos)
    extractNeg<<-intersect(names(finalWords) , neg)

    # counting means, medians and sums of the proportion
positive/negative
    meanStars<-mean(colSums(finalWords[,extractPos]))/mea
n(colSums(finalWords[,extractNeg]))
    medianStars<-median(colSums(finalWords[,extractPos]))
/median(colSums(finalWords[,extractNeg]))
    sumStars<-sum(colSums(finalWords[,extractPos]))/sum(c
olSums(finalWords[,extractNeg]))

    # extracting positive and negative words from all fin
al words
    extract<<-finalWords[,union(extractPos,extractNeg)]
    c(meanStars,medianStars,sumStars)
```

The result of the proportion function is used for the building of the most frequently used clouds of words [4, 5] (corpuses of words, to be more exact), corresponding to the frequency diagrams for each of 5 possible ratings.

III. Histograms and Clouds

Figures 1-5 illustrate the histograms and clouds for each of 5 possible meanings of the rating.

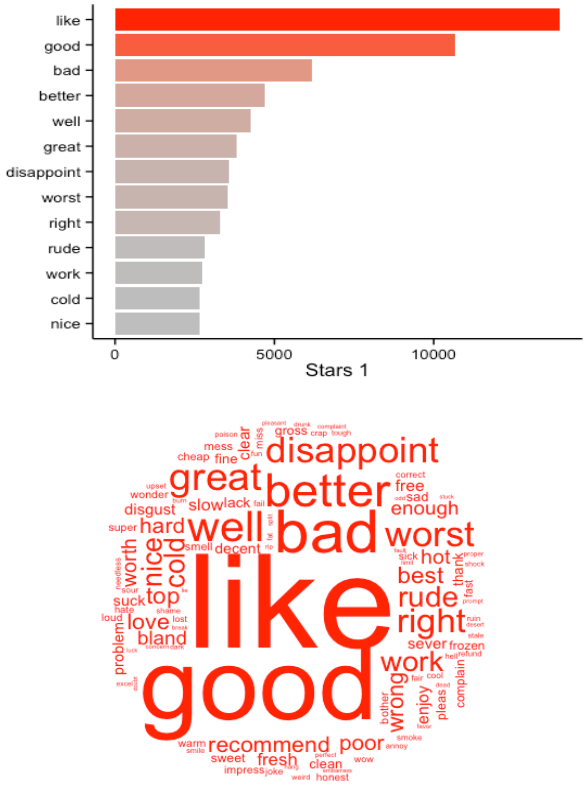

Figure 1. The most common words in the 1-star ratings: the histogram with frequency and the clouds with the words

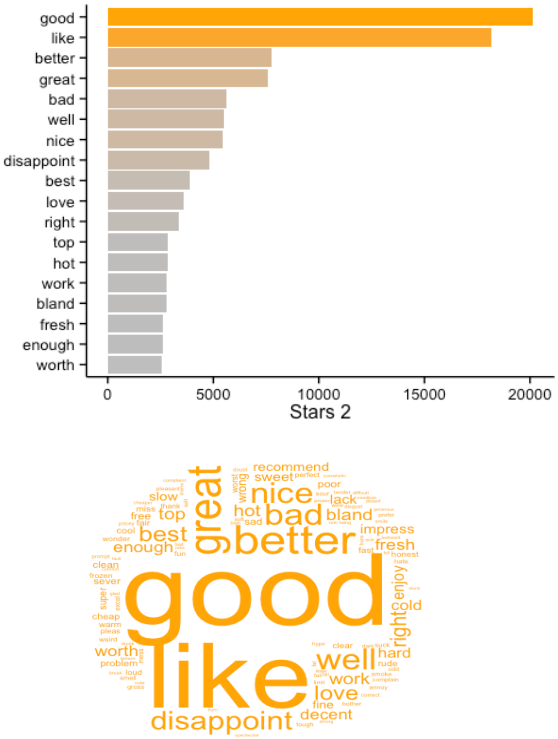

Figure 2. The most common words in the 2-star ratings: the histogram with frequency and the clouds with the words

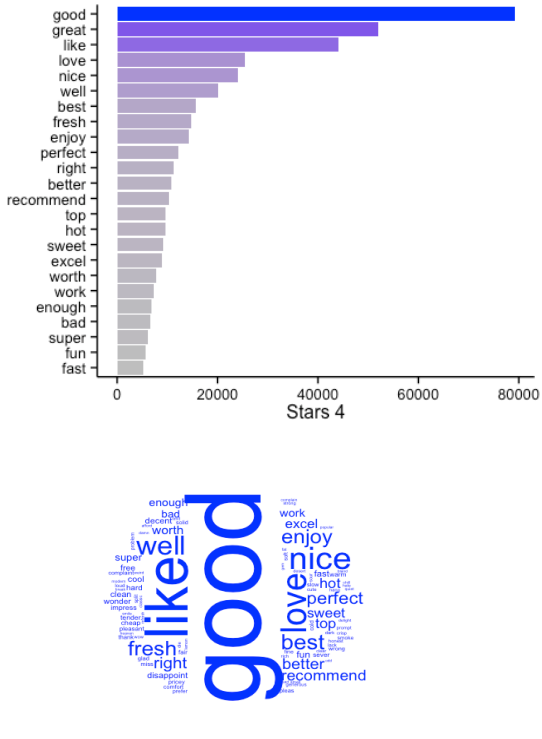

Figure 4. The most common words in the 4-star ratings: the histogram with frequency and the clouds with the words

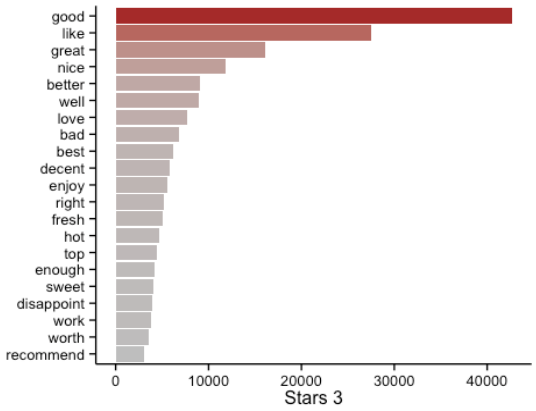

Figure 3. The most common words in the 3-star ratings: the histogram with frequency and the clouds with the words

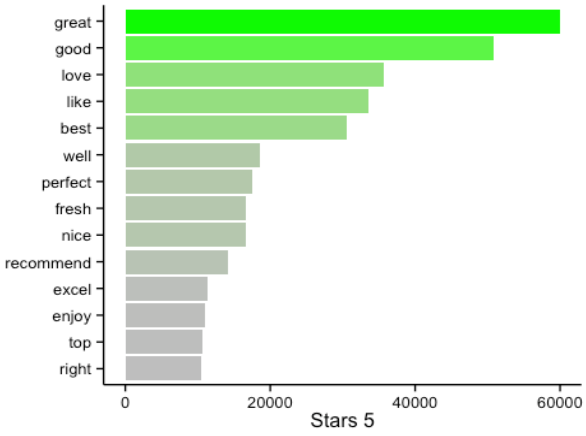

Figure 5. The most common words in the 5-star ratings: the histogram with frequency and the clouds with the words

IV. Results of the Analysis

The analysis means, medians and sums of the proportion positive/negative:

```
prop<-as.data.frame(s)
rownames(prop)<-c("mean","median","sum")
colnames(prop)<-c("Star1","Star2","Star3","Star4","Star5"
)
prop


##           Star1    Star2    Star3    Star4    Star5
## mean   1.921408 2.695613 3.247577 3.968369 3.829088
## median 2.024209 2.136704 2.508353 2.090584 1.937760
## sum    1.372434 2.190185 3.844071 6.689536 9.920819
```

Figure 6 demonstrates the dependence of the quantity of stars in the rating on the number of the clients' reviews.

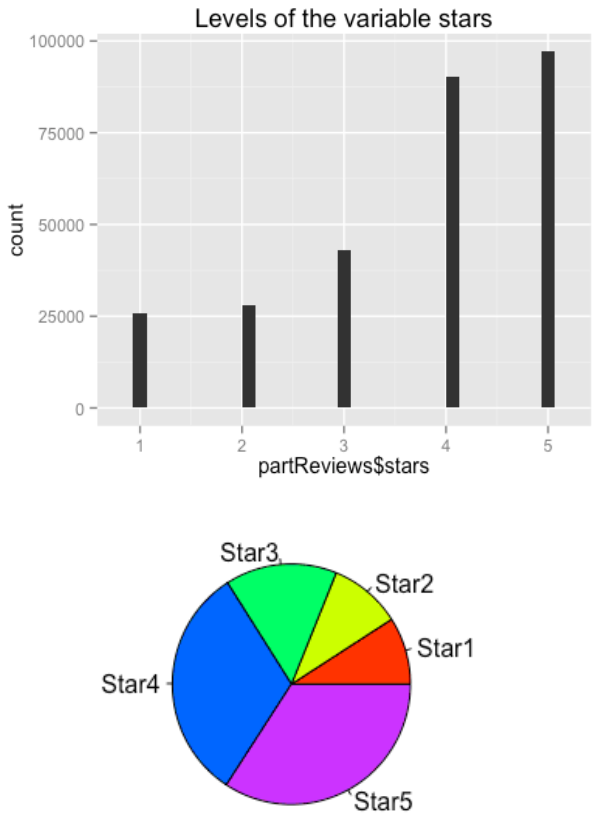

Figure 6. The quantity of levels "Star1","Star2","Star3","Star4","Star5" in the partReviews dataset for variable "stars"

Figure 7 graphically reflects the results of the last line of the above mentioned table.

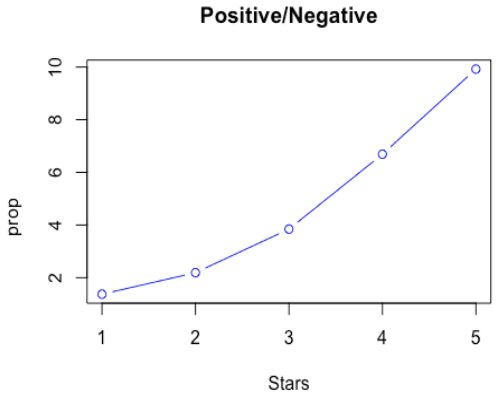

Figure 7. The total sums of the proportions of the positive and negative words

In Figure 7 we mean that:

```
prop<-
sum(colSums(finalWords[,extractPos]))/sum(colSums(finalWo
rds[,extractNeg]))
```

V. Discussion

Having worked with the several groups of random samples we propose to discuss this result:

```
if stars=1 then prop  ≤  1.5
if stars=2 then 1.5 < prop ≤ 3
if stars=3 then 3< prop ≤ 5.5
if stars=4 then  5.5< prop ≤ 8.5
if stars=5 then  8.5< prop
```

This is our interpretation of the results of the analysis for our question/problem.

VI. Conclusions

If you look at any business, associated with restaurants and bars, it is possible to estimate the value prop = sum(colSums(finalWords[,extractPos]))/sum(colSums(finalWords[,extractNeg])) from the count of stars in the feedbacks from the visitors and vice versa, knowing the value prop, it is possible to predict the number of stars in the rating.

References

[1] H. Tang, S. Tan, X. Cheng, A survey on sentiment detection of reviews, Expert Systems with Applications 36 (7) (2009) 10760-10773.

[2] Walaa Medhat, Ahmed Hassanb, Hoda Korashy, Sentiment analysis algorithms and applications: A survey, Ain Shams Engineering Journal, Volume 5, Issue 4, (2014) 1093–1113.

[3] Information on http://fotiad.is/blog/sentiment-analysis-comparison/

[4] Information on http://thinktostart.com/create-twitter-sentiment-word-cloud-in-r/

[5] Information on http://www.analyticsvidhya.com/blog/2014/05/build-word-cloud-text-mining-tools/

[6] Information on https://www.kaggle.com/yelp-dataset/yelp-dataset/