International Journal of Research in Engineering, Science and Management
Volume-3, Issue-8, August-2020
journals.resaim.com/ijresm | ISSN (Online): 2581-5792 | RESAIM Publishers

Prediction of Cell Phone Sales from Online Reviews Using Text Mining

Trishal Jadhav*
Student, Department of Electrical and Electronics Engineering, RV College of Engineering, Bengaluru, India
*Corresponding author: trishaljadhav@gmail.com

***Abstract*: In a world that is constantly shifting more and more towards a digital realm, having data available for most fields is becoming all the easier. Since all online interactions are automatically recorded, they can be readily accessed to provide valuable data. This helps in tracing and predicting consumer behavior which can help predict the commercial success of various day to day products. One way to obtain usable data from online sources is by using text mining which is a data science technique which analyses textual data and sifts through it to obtain valuable chunks which can be used as an insight without having to convert it to numerical format. Text mining can be done using various tools such as Python and R and further operations can be performed on the data to produce more valuable insights. This paper describes a technique of text analytics on mobile online reviews to discover significant text patterns that exist in the documents. These documents are considered as unstructured textual data. The paper proposes a framework that consists of 3 stages (i) data collection (ii) document preprocessing (iii) text analytics and visualization and (iv) prediction by comparison. The technique is developed using R text mining package for text analytics experiments. We use the patterns obtained from this textual data to try and predict the commercial success of a new cell phone model by using its online review and calculating similarity with reviews of commercially successful cell phones.**

***Keywords*: Data analytics, Data science, Textual data, Text mining.**

1. Introduction

Often when we are required to obtain outcomes from data, we need it to be quantitative in form. When data is in quantitative form it is easier to work with and can be directly plugged into existing algorithms to obtain meaningful insights from the data. However, most data are textual in form so there arises a problem.

Textual data is more wholesome and meaningful when compared to quantitative data. Usually to extract meaning from textual data we convert it to certain measurable parameters and then plug these into known algorithms. However, when we do this, we lose some of the most important pieces of the data. Most importantly, the sentiment that was meant to be delivered through the text is lost in the process and this in turn is a hug loss for anyone trying to obtain meaningful insights from the text.

This is where text mining comes into place. We can use text mining to obtain meaningful information from structured and unstructured textual data to be able to process it and form insights.

In this experiment we obtain frequency of occurrence of words from online reviews of cell phones and use this data to predict the commercial success of a new cell phone model in the Indian market. We obtain the term document matrix of the reviews of the top 7 selling cell phone models in India and compare them to reviews of new cell phones and calculate similarity of most commonly used words to check whether the new model will be successful or not.

2. Literature Survey

This section deals with an extensive survey and provides a guide and background for the entire work. It mainly includes the review on the earlier techniques practiced before the uprising of artificial intelligence techniques.

Sanjiv R. Das [1] discusses about the current landscape of text analytics in finance. He examines how text is extracted from various web sites and services using R. He also deals with the basics of text analytics such as dictionaries, lexicons, mood scoring, and summarization of text. This is followed by the analytics of text classification. Finally, the he takes a look at the future of text analytics.

Gary Miner et. al [2] gives a comprehensive reference on how to conduct text mining and statistically analyze results in his book. In addition to providing an in-depth examination of core text mining, they examine advanced preprocessing techniques and visualization approaches. Finally, they explore current real-world, mission-critical applications of text mining using real world example tutorials in such varied fields as corporate, finance, business intelligence, genomics research, and counterterrorism activities.

Zuraini Zainol et. al [3] gives an account of the techniques of text analytics on peace-keeping documents to discover significant text patterns exist in the documents in his paper. These documents are considered as unstructured textual data.

He proposed a framework that consists of 3 stages:

1) Data collection
2) Document preprocessing and
3) Text analytics and visualization.

The technique is developed using R text mining package for text analytics experiments.

Arman Khadjeh Nassirtouss et. al [4] discusses a novel approach to predict directional-movements of a currency-pair in the foreign exchange market based on the text of breaking financial news-headlines. The motivation is twofold: First, although market-prediction through text-mining is shown to be a promising effort, the text-mining approaches utilized in it at this stage are basic ones as it is still an emerging field. The second part is to research the foreign exchange market, which seems not to have been researched using predictive text-mining.

3. Motivation

As seen in the literature section there has been a rising trend in using text mining to obtain data without losing its qualitative features and using this data to predict outcomes of events. Any time a text document is readily available on a topic, it can be mined to obtain predominantly opinionated data.

One such application is the prediction of commercial success of a product using the reviews and reactions of articles published at the time of its release. This can be associated to automobiles, clothing, electronics, and several other articles that people research before purchasing. This study deals with cell phones as the field of research, as their range of prices is comparatively low, and they have an abundance of literature on their individual performance and appeal.

4. Objectives

The objectives of the project are:

- To collect the relevant data to derive clear results.
- To perform and verify the methodology of using text mining in R.
- Visualizing the outputs of the model.
- Obtaining qualitative results and conclusions by comparison.

5. Data Used

The data used for this experiment were derived from textual documents of online mobile reviews on

https://www.gadgetsnow.com/. The top 7 bestselling cell phones in India in the year 2019 were determined using data on the website as follows:

- Realme 3 Pro
- Realme 3
- Realme 5
- Redmi 7A
- Redmi Note 7S
- Samsung Galaxy M20
- Vivo Z1 Pro

The critical reviews of these individual phones were taken from the website and stored as text files. To test the model, the phone “Vivi X50 Pro” was considered and its review was also stored.

6. Experimental Work

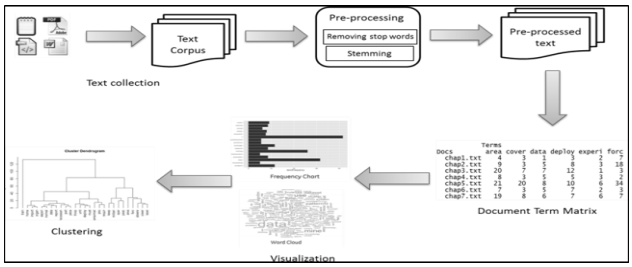

Fig. 1. Methodology of the experiment

A. *Text Collection*

First, we require the data from which we are needed to infer some information from. In this case we need the text document containing the reviews of the cell phones. We can either read the documents directly from the webpage using R function *readLines()* or we can store them in a text document such as a word document or preferably on NotePad so it can be processed more easily [1]. We took a list of the top 7 selling cell phone models in India and collected their reviews in one NotePad file. We read the text after loading libraries shown in Fig. 2 using the code in Fig. 3.

```
1  library(data.table)
2  library(dplyr)
3  library(ggplot2)
4  library(reshape2)
5  library(tm)
6  library(XML)
7  library(e1071)
8  library(stringr)
9  library(wordcloud)
10 library(SnowballC)
11 library(RColorBrewer)
```

Fig. 2. Loading required R libraries

```
13 setwd("~/Downloads/Text Mining Paper")
14
15 #Loading text documents#
16 text1 <- readLines("Top Selling Phones.rtf")
17 docs <- Corpus(VectorSource(text1))
18 inspect(docs)
```

Fig. 3. R code to read text files and create a text corpus

B. *Pre-Processing of Text*

After we obtain this document, we need to pre-process the text so that we can obtain data free of random variable and which are more meaningful. We use the text mining package *tm()* in R to perform most of the operations in this experiment as seen in Fig. 4.

First, we convert our text documents into a text corpus which is a consolidated structure of text which makes it easier for us to use this data to work with. After converting this data into a text corpus, we clean it up by removing unnecessary spaces and characters such as “/”, “- “,”.”,”;”,” []”,” {}”,” ()”,” %”,” &”,” \”,” @”,” #”,” $” and “?” as well as brand names which may

lead to a bias [1]. In Fig. 3 we see the text corpus before cleaning, and we use the code in Fig. 4 to obtain the clean text corpus seen in Fig. 5.

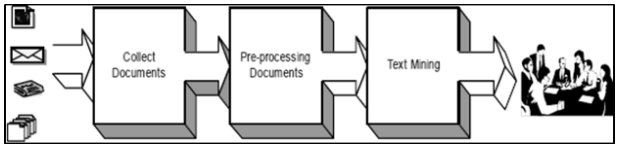

Fig. 4. Flow of the text document processing

```
[11] If two\\'92s company, three\\'92s a crowd then we don\\'92t
ny brands jostling for space and to get the customers\\'92 attent
 a name for itself. However, to its credit Chinese brand Realme h
 offerings. The latest smartphone launched by the brand is Realme
t smartphone for a while and here\\'92s our review: When it comes
tiate. Even more so, if the phone in question is a successor to a
ro but has a few differences. The back panel is slightly more cur
tic yet doesn\\'92t look too much of a budget smartphone. A lot o
 \\'97 our review unit was this one \\'97 and there are Carbon Gr
 fits easily in the pockets as well. You\\'92ll find a 3.5mm head
ealme 3 Pro has a SIM tray which can fit in two nano cards and a
ensor at the back. Unlike a few other devices, the cameras don\\'
tch for the front camera and the earpiece. Speaking of display, t
er, the chin at the bottom is more than noticeable. The full HD+
olours are produced fine with good viewing angles as well. There
ct sunlight. As per the company, the real star of the show in the
ssor and is at par with what rivals offer. For instance, Xiaomi R
s to offer better graphics handling capabilities. Also, it suppos
 found that on both the above counts, Realme 3 Pro delivered resp
nd 6GB RAM + 128GB of storage. Our review unit was the latter hig
 the Realme 3 Pro except one or two times when certain heavy-duty
```

Fig. 5. Text corpus before cleaning

```
#Cleaning Text Document#
toSpace <- content_transformer(function(x,pattern)gsub(pattern," ",x))
docs <- tm_map(docs,toSpace,"/")
docs <- tm_map(docs,toSpace,"@")
docs <- tm_map(docs,toSpace,"\\|")
docs <- tm_map(docs,content_transformer(tolower))
docs <- tm_map(docs,removeNumbers)
docs <- tm_map(docs,removeWords,stopwords("english"))
docs <- tm_map(docs,removeWords,c("deftab","pardpardeftabrisaqjpartightenfactor","ffs","fs","cf",
                                  "pardpardeftabrjpartightenfactor",
                                  "rtfansiansicpgcocoartfcocoasubrtf",
                                  "pardpardeftabrisapartightenfactor",
                                  "paperwpaperhmarglmargrviewwviewkind",
                                  "fonttblffswissfcharset arialmtffnilfcharset calibri",
                                  "colortblredgreenblueredgreenblue","expandedcolortblcssrgbcccc"))
docs <- tm_map(docs,removePunctuation)
docs <- tm_map(docs,stripWhitespace)
docs <- tm_map(docs,removeWords,c("realme","redmi","xiaomi","samsung","can","comes","device","smartphone",
                                  "mode","note","pro","phone","camera","battery","display","also","get","vivo","will","screen"))
```

Fig. 6. R code for cleaning text corpus

```
[11]  twos company threes crowd dont know describe budget  market
 hard brand come make name however credit chinese brand  managed
 challenge market leader  used budget  s review  design stage bud
  looks quite similar  differences back panel slightly curved bl
 options company given s nitro blue review unit one carbon grey l
 headphone jacket microusb port single speaker bottom  sim tray
e devices cameras dont protrude tucked neatly back  s notch front
ble full hd  resolution makes good enough  watch multimedia conte
n direct sunlight per company real star show  processor  runs qua
ul processor claims offer better graphics handling capabilities
rmance  two variants gb ram gb storage gb ram gb storage review
e two times certain heavyduty files took longer share budget user
 slow performing graphicintensive tasks much surprise found singl
stom uis  runs coloros  customised host preloaded apps come  me
irdrop  users  share filed  features dual  set mp sony imx senso
 aspect  adequate job  broad daylight clicked really good imag
as images sharp fact detailed  comparison  managed hold quit
owlight photography found  delivered competent results s nightsc
esults quite good s mp selfie  manages click good images daylig
urprisingly images looked really artificial  shoots videos unto
```

Fig. 7. Text corpus after cleaning

C. *Creating a term document matrix*

After cleaning the textual data we can use it to create a term document matrix using the *TermDocumentMatrix()* function of the *tm* package in R.

A term document matrix is a matrix that is used to depict the frequency of occurrence of words in the document. We sort the matrix so that we get the most frequently occurring words at the top and the *head()* function gives us the first few lines of the matrix as shown in Fig. 5 and use *inspect()* to obtain the properties shown in Fig. 6.

```
#Building Term Document Matrix#
dtm <- TermDocumentMatrix(docs)
m <- as.matrix(dtm)
v <- sort(rowSums(m),decreasing = TRUE)
d <- data.frame(word = names(v),freq = v)
head(d,20)
inspect(dtm)
```

Fig. 8. R code to build term document matrix

	word	freq
good	good	57
one	one	42
like	like	41
sensor	sensor	34
performance	performance	32
features	features	29
images	images	28
price	price	28
two	two	28
well	well	28
colour	colour	25
however	however	23
apps	apps	22
back	back	22
front	front	21
shots	shots	21
storage	storage	21
face	face	21
design	design	20
smartphones	smartphones	20

Fig. 9. First 20 lines of term document matrix

```
> inspect(dtm)
<<TermDocumentMatrix (terms: 1695, documents: 27)>>
Non-/sparse entries: 3169/42596
Sparsity           : 93%
Maximal term length: 41
Weighting          : term frequency (tf)
Sample             :
             Docs
Terms         1 11 13 15 17 19 2 21 23 9
  features    0  2  6  5  5  1 0  1  9 0
  good        0  8  6  9  3  1 0  9 21 0
  images      0  6  0  7  7  5 0  0  3 0
  like        0  2 10  6 10  4 0  4  5 0
  one         0  3 11  8  3  3 0  3 11 0
  performance 0  6  8  2  1  3 0  3  9 0
  price       0  2  1  6  4  7 0  5  3 0
  sensor      0  3  8  3  3 11 0  2  4 0
  two         0  3  7  6  4  4 0  3  1 0
  well        0  6  4  1  3  2 0  2 10 0
```

Fig. 10. Inspecting term document matrix

D. *Visualizing the data*

The term document matrix gives us a rough idea of the most frequently occurring words in a text document. However, it may not be the easiest way to digest the data. For this reason, we use other visualizing methods such as wordcloud and frequency charts to make it more visually friendly. To use wordcloud we load the *wordcloud* library and use the *wordcloud()* function as shown in Fig. 7. We use colors from the *RColorBrewer* library to fill the wordcloud with colors [3] and the result is seen in Fig 8. To create the frequency chart we use *barplot()* function from the *ggplot2* libraryas seen in Fig. 9. The frequency chart

simulated is shown below in Fig. 10.

```
#Generating Word Cloud#
set.seed(1234)
wordcloud(words = d$word, freq = d$freq, min.freq = 1, max.words = 200, random.order = FALSE,
          rot.per = 0.35, colors = brewer.pal(8,"Dark2"))
```

Fig. 11. R code to generate a word cloud

Fig. 12. Wordcloud of term document matrix

```
#Generating Bar Plot#
barplot(d[2:20,]$freq, las = 2, names.arg = d[2:20,]$word, col = "lightblue",
        main = "Most frequent words",ylab = "Word Frequencies")
```

Fig. 13. R code to generate a frequency bar plot

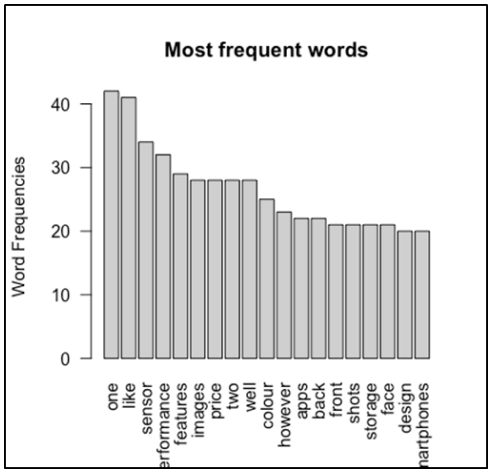

Fig. 14. Frequency chart of the term document matrix

E. Clustering

Hierarchal clustering is a method of segregating the words found in the document using their frequency [3]. They are segregated into more than branches using Euclidian distancing. We can perform clustering on R by using the *dist()* and *hclust()* function.

```
#Clustering#
di <- head(d,20)
dt <- dist(di,method = "euclidean")
clu <- hclust(dt,method = "complete")
plot(clu,cex = 0.6,hang = -1)
```

Fig. 15. R code for clustering

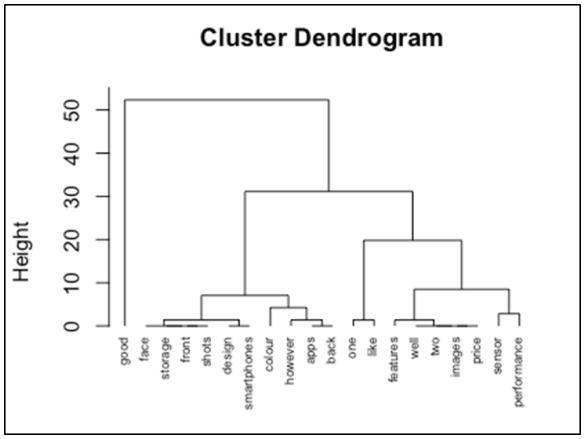

Fig. 16. Hierarchal clustering of words in term document matrix

7. Results and Observations

After performing the analysis on the text corpus containing the reviews of commercially successful phone, we can determine the likelihood of success of a cell phone model by comparing the term document matrix of it's review with that of the successful phones.

To do so, the "Vivi X50 Pro" which was released in July 2020 was chosen and its critical review was converted into a text corpus. The wordcloud, frequency chart and clustering for this text corpus is shown in Fig. 17, Fig. 18 and Fig. 19 respectively.

Fig. 17. Wordcloud of the test document

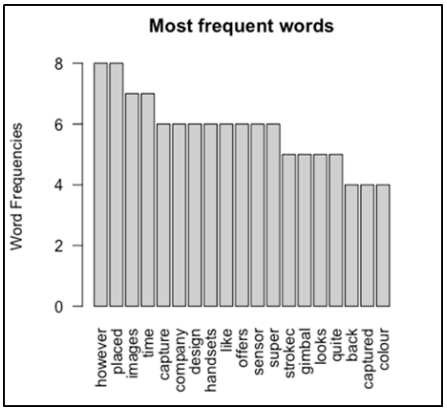

Fig. 18. Frequency chart of the test document

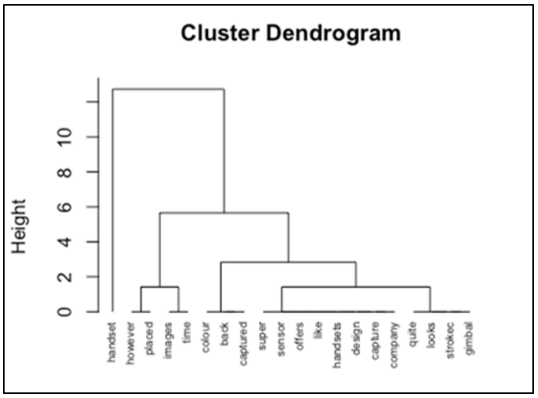

Fig. 19. Clustering of the test document

The similarity between the two text corpuses was derived by using a simple formula where the number of common words is divided by the total number of unique words in the two documents as shown in Fig. 20.

```
446  mat1 <- as.data.frame(as.matrix(dtm))
447  mat2 <- as.data.frame(as.matrix(dtm_9))
448  all_words <- unique(c(rownames(mat1),rownames(mat2)))
449  common_words <- intersect(rownames(mat1),rownames(mat2))
450  sim_index <- length(common_words)/length(all_words)
451  print(paste0("Similarity Index = ",round((sim_index*100),digits = 2),"%"))
```

Fig. 20. R code to calculate similarity index

The similarity index was calculated as 17.32% between the test review and the text corpus containing the combined reviews of all top 7 phones. The similarity between the test review and the individual reviews of all top 7 phones was also calculated and is displayed as a similarity vector as shown in Fig. 21.

```
[1] "Similarity Index = 17.32%"
> similarity_vector <- c(sim_index,sim_index_2,sim_index_3,sim_index_4,sim_index_5,sim_index_6,sim_index_7,sim_index_8)
> similarity_vector
[1] 0.1732326 0.1731044 0.1885880 0.2056338 0.1797753 0.2393398 0.1537335 0.2029915
```

Fig. 21. Displaying similarity index and similarity vector

8. Conclusion

After performing the experiment, we arrived at the common words used in the online reviews of these commercially successful phones. We then compared reviews of a new cell phone and checked if they have a similar composition of these words. The similarity was calculated as an average of 19.19%. After comparison with previously recorded data, it was found that most commercially successfully phones have an average similarity index with the top 7 reviews of 15% or more. Hence, we can make a safe prediction that the phone “Vivi X50 Pro” has a good chance of bring commercially successful.

References

[1] Sanjiv R Das, Santa Clara University, “Text and Context: Language Analytics in Finance”, Now Publishers, 2014. ISBN: 978-1-60198-910-9.

[2] Gary Miner, Dursun Delen, John Elder, Andrew Fast, Thomas Hill, Robert A. Nisbet, “Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications”, Elsevier Inc., 2012. ISBN: 978-0-12-386979-1.

[3] Zuraini Zainol, Puteri N.E. Nohuddin, Tengku A.T. Mohammed and Omar Zakaria, “Text Analytics of Unstructured Textual Data: A Study on Military Peacekeeping Document Using R Text Mining Package”, 2017, ICOCI.

[4] Arman Khadjeh Nassirtouss, Saeed Aghabozorgi, TehYing Wah, David Chek Ling Ngo, “Text mining of news-headlines for FOREX market prediction: A Multi-layer Dimension Reduction Algorithm with Semantics and Sentiment”, 2014, Elsevier Inc.