ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 7, Issue 9, September 2021

Text Mining Of Jane Austen Novel "Pride and Prejudice"

Palli Jhansi[1], V Aanjali Priyaka[2], D Murali Manohar[3], Naveen Kumar Jeena[4], Satya Ravindra Babu R[5]

U.G. Scholars, Department of CSE, Sanketika Vidya Parishad Engineering College, Visakhapatnam [1,2,3,4]

Assistant Professor, Department of CSE, Sanketika Vidya Parishad Engineering College, Visakhapatnam [5]

Abstract— Text mining, also referred to as *text data mining*, similar to text analytics, is the process of deriving high-quality information from text. In this study, the authors thoroughly perform text mining on a Jane Austin Novel "Pride And Prejudice" as a case study. Firstly, we replace "/", "@" and "|" with space. We then convert the text to lower case. We remove numbers and common English stopwords. We also remove punctuations and eliminate extra white spaces. We then do Text Stemming which reduces the word to its root form. We then build a Term Document Matrix, sort by decreasing order of frequency and display the 500 most frequent words. We then plot the 100 most frequent words. We finally generate a Word Cloud for a maximum of 500 words.

***Index Terms*— Text Mining**

I. INTRODUCTION

With the advancement of technology, more and more data is available in digital form. Among which, most of the data (approx. 85%) is in unstructured textual form. Text, so it has become essential to develop better techniques and algorithms to extract useful and interesting information from this large amount of textual data. Hence, the area of text mining and information extraction has become popular areas of research, to extract interesting and useful information.

In general Text mining consists of the analysis of text documents by extracting key phrases, concepts, etc. and prepare the text processed for further analyses with data mining techniques. This paper, discussed the concept, process and applications of text mining, which can be applied in multitude areas such as webmining, medical, resume filteration, etc. It also enlighten the hidden potential that lies in the field of text mining and motivated to explore it further. Text mining is defined as —the non-trivial extraction of hidden, previously unknown, and potentially useful information from (large amount of) textual data'' [1]. Text Mining is a new field that tries to extract meaningful information from natural language text. It can be defined as the process of analyzing text to extract information that is useful for a specific purpose. Compared with the type of data stored in databases, text is unstructured, ambiguous, and difficult to process. Nevertheless, in modern culture, text is the most communal way for the formal exchange of information. Text mining usually deals with texts whose function is the communication of actual information or opinions, and the stimuli for trying to extract information from such text automatically is fascinating - even if success is only partial.

Text mining is similar to data mining, except that data mining tools [2] are designed to handle structured data from databases, but text mining can also work with unstructured or semi-structured data sets such as emails, text documents and HTML files etc. As a result, text mining is a far better solution.

Text mining usually is the process of structuring the input text (usually parsing, along with the addition of some derived linguistic features and the removal of others, and subsequent insertion into a database), deriving patterns within the structured data, and final evaluation and interpretation of the output.

The term — text mining is commonly used to denote any system that analyzes large quantities of natural language text and detects lexical or linguistic usage patterns in an attempt to extract probably useful (although only probably correct) information.

II. TEXT MINING USING R

Text Mining Using R

Installing and loading R packages

The following packages are used in this project:

- **tm** for text mining operations like removing numbers, special characters, punctuations and stop words (Stop words in any language are the most commonly occurring words that have very little value for NLP and should be filtered out. Examples of stop words in English are "the", "is", "are".)
- **snowballc** for stemming, which is the process of reducing words to their base or root form. For example, a stemming algorithm would reduce the words "fishing", "fished" and "fisher" to the stem "fish".
- **wordcloud** for generating the word cloud plot.
- **RColorBrewer** for color palettes used in various plots
- **ggplot2** for plotting graphs

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 7, Issue 9, September 2021

Reading file data into R

The R base function read.table() is generally used to read a file in table format and imports data as a data frame. Several variants of this function are available, for importing different file formats;

- **read.csv() is** used for reading comma-separated value (csv) files, where a comma "," is used a field separator
- **read.delim()** is used for reading tab-separated values (.txt) files

The input file has multiple lines of text and no columns/fields (data is not tabular), so we use the readLines function. This function takes a file (or URL) as input and returns a vector containing as many elements as the number of lines in the file. The readLines function simply extracts the text from its input source and returns each line as a character string. The n= argument is useful to read a limited number (subset) of lines from the input source (Its default value is -1, which reads all lines from the input source). When using the filename in this function's argument, R assumes the file is in our current working directory (we can use the getwd() function in R console to find our current working directory). We can also choose the input file interactively, using the file.choose() function within the argument. The next step is to load that Vector as a Corpus. In R, a Corpus is a collection of text document(s) to apply text mining or NLP routines on. Details of using the readLines function are sourced from: https://www.stat.berkeley.edu/~spector/s133/Read.html

Cleaning up Text Data

Cleaning the text data starts with making transformations like removing special characters from the text. This is done using the tm_map() function to replace special characters like /, @ and | with a space. The next step is to remove the unnecessary whitespace and convert the text to lower case.

We then remove the *stopwords*. They are the most commonly occurring words in a language and have very little value in terms of gaining useful information. They should be removed before performing further analysis. Examples of stopwords in English are "the, is, at, on". There is no single universal list of stop words used by all NLP tools. stopwords in the tm_map() function supports several languages like English, French, German, Italian, and Spanish. One should note that the language names are case sensitive. Next, we remove numbers and punctuation.

The last step is text stemming. It is the process of reducing the word to its root form. The stemming process simplifies the word to its common origin. For example, the stemming process reduces the words "fishing", "fished" and "fisher" to its stem "fish". It should be noted that stemming uses the *SnowballC* package.

Building the term document matrix

After cleaning the text data, the next step is to count the occurrence of each word, to identify popular or trending topics. Using the function TermDocumentMatrix() from the text mining package, we can build a Document Matrix – a table containing the frequency of words.

Plotting the top 5 most frequent words using a bar chart is a good basic way to visualize this word frequent data.

Generating the Word Cloud

A word cloud is one of the most popular ways to visualize and analyze qualitative data. It's an image composed of keywords found within a body of text, where the size of each word indicates its frequency in that body of text. We use the word frequency data frame (table) created previously to generate the word cloud.

Below is a brief description of the arguments used in the word cloud function;

- **words** – words to be plotted
- **freq** – frequencies of words
- **min.freq** – words whose frequency is at or above this threshold value is plotted (in this case, I have set it to 5)
- **max.words** – the maximum number of words to display on the plot (in the code above, I have set it 100)
- **random.order** – I have set it to FALSE, so the words are plotted in order of decreasing frequency
- **rot.per** – the percentage of words that are displayed as vertical text (with 90-degree rotation). I have set it 0.40 (40 %), please feel free to adjust this setting to suit our preferences
- **colors** – changes word colors going from lowest to highest frequencies

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 7, Issue 9, September 2021

III. R PROGRAM FOR TEXT MINING

R Program For Text Mining

```
# Install
install.packages("tm")  # for text mining
install.packages("SnowballC") # for text stemming
install.packages("wordcloud") # word-cloud generator
install.packages("RColorBrewer") # color palettes
install.packages("ggplot2") # for plotting graphs
# Load
library("tm")
library("SnowballC")
library("wordcloud")
library("RColorBrewer")
library("ggplot2")

# Read the text file from local machine , choose file interactively
text <- read.delim(file.choose())
#text <- readLines(file.choose())
# Load the data as a corpus
TextDoc <- Corpus(VectorSource(text))

#Replacing "/", "@" and "|" with space
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
TextDoc <- tm_map(TextDoc, toSpace, "/")
TextDoc <- tm_map(TextDoc, toSpace, "@")
TextDoc <- tm_map(TextDoc, toSpace, "\\|")
# Convert the text to lower case
TextDoc <- tm_map(TextDoc, content_transformer(tolower))
# Remove numbers
TextDoc <- tm_map(TextDoc, removeNumbers)
# Remove english common stopwords
TextDoc <- tm_map(TextDoc, removeWords, stopwords("english"))
# Remove your own stop word
# specify your custom stopwords as a character vector
TextDoc <- tm_map(TextDoc, removeWords, c("s", "company", "team"))
# Remove punctuations
TextDoc <- tm_map(TextDoc, removePunctuation)
# Eliminate extra white spaces
TextDoc <- tm_map(TextDoc, stripWhitespace)
# Text stemming - which reduces words to their root form
TextDoc <- tm_map(TextDoc, stemDocument)

# Build a term-document matrix
TextDoc_dtm <- TermDocumentMatrix(TextDoc)
dtm_m <- as.matrix(TextDoc_dtm)
# Sort by descearing value of frequency
dtm_v <- sort(rowSums(dtm_m),decreasing=TRUE)
dtm_d <- data.frame(word = names(dtm_v),freq=dtm_v)
# Display the top 100 most frequent words
head(dtm_d, 100)

# Plot the most frequent words
barplot(dtm_d[1:25,]$freq, las = 2, names.arg = dtm_d[1:25,]$word,
        col ="lightgreen", main ="Top 25 most frequent words",
        ylab = "Word frequencies")

#generate word cloud
set.seed(1234)
wordcloud(words = dtm_d$word, freq = dtm_d$freq, min.freq = 5,
          max.words=500, random.order=FALSE, rot.per=0.40,
          colors=brewer.pal(8, "Dark2"))
```

IV. RESULTS

The Results and Conclusion (Frequencies Of The 100 Most Frequent Words & Word Cloud) of Text Mining Of the Jana Austen Novel 'Pride And Prejudice' are detailed below:

Frequencies Of The 100 Most Frequent Words

Table 1 – Frequencies of the 100 most frequent words

	word	freq
elizabeth	elizabeth	497
will	will	396
said	said	372
darci	darci	368
bennet	bennet	309
mrs	mrs	302
much	much	300
must	must	296
bingley	bingley	288
one	one	270
sister	sister	265
miss	miss	263
jane	jane	260
everi	everi	251
know	know	248
ladi	ladi	233
think	think	221
never	never	211
though	though	206
time	time	206
soon	soon	199
now	now	198
can	can	195
see	see	195

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 7, Issue 9, September 2021

say	say	193
well	well	192
make	make	189
might	might	185
may	may	175
wish	wish	175
good	good	169
wickham	wickham	168
thing	thing	167
littl	littl	165
day	day	163
hope	hope	161
noth	noth	160
without	without	159
collin	collin	158
look	look	158
dear	dear	156
shall	shall	156
come	come	152
friend	friend	152
give	give	150
even	even	149
lydia	lydia	146
great	great	145
like	like	145
feel	feel	143
famili	famili	140
happi	happi	139
man	man	139
manner	manner	130
believ	believ	128
first	first	128
mother	mother	126
howev	howev	125
two	two	125
young	young	125
letter	letter	123
thought	thought	123
daughter	daughter	121
father	father	120
long	long	120
repli	repli	120
ever	ever	119
certain	certain	116
made	made	116
last	last	115
catherin	catherin	113
quit	quit	112
walk	walk	110
marri	marri	109
room	room	108
mani	mani	105
alway	alway	104
away	away	104
expect	expect	102
mean	mean	102
return	return	102
talk	talk	102
love	love	98
way	way	98
enough	enough	97
receiv	receiv	97
speak	speak	96
sure	sure	96
attent	attent	95
saw	saw	95
hous	hous	94
seem	seem	94
take	take	93
appear	appear	92
cri	cri	92
felt	felt	92
answer	answer	91
gardin	gardin	89
hear	hear	89
kind	kind	89

Plot Of Top 25 Most Frequent Words

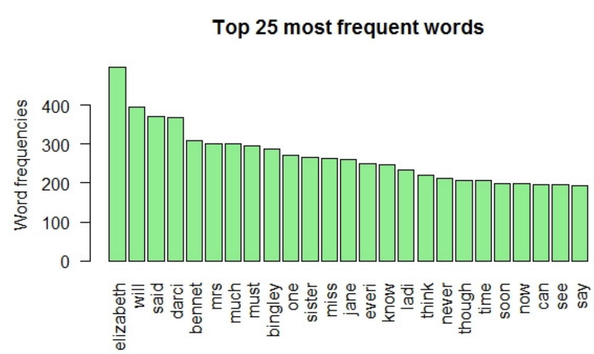

Fig 1 – Plot of the Top 25 Most Frequent Words

Word Cloud

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 7, Issue 9, September 2021

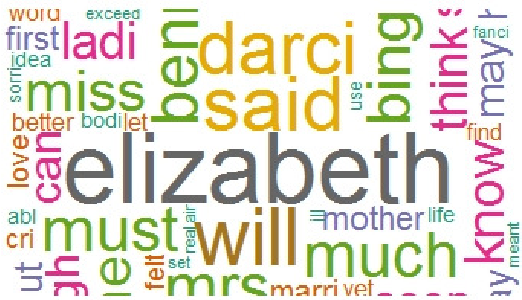

Fig 2 – Word Cloud showing the Most Frequent Words

V. CONCLUSION

The Pros of a Wordcloud are:

1) *It reveals the essential.*
2) *They delight and provide emotional connection.* Both the creation of a word-cloud and the observation of one help to provide an overall sense of the text. The same visceral response doesn't happen when staring at a page of text.
3) *They're fast.* Poring over text to develop themes from research takes time.
4) *They're engaging.* Visual representation of data tends to have an impact and generates interest amongst the audience. Word clouds can allow you to share back results from research in a way that doesn't require an understanding of the technicalities.

The Cons of a Wordcloud are:

Size isn't everything. Although the Word Cloud is designed to make words stand out according to their size based on their frequency of occurrence, other factors can affect the visual 'decoding' of the data from the observer's perspective. For example, the length of the word and the white space around the glyphs (letters) can make it look more or less important relative to others in the cloud. This can mislead your interpretation.

REFERENCES

[1] Daniel Waegel. ―The Development of Text-Mining Tools and Algorithms‖. Ursinus College, 2006.
[2] Navathe, Shamkant B. and Elmasri Ramez. ―Data Warehousing and Data Mining‖, in ―Fundamentals of Database Systems‖, Pearson Education pvt Inc, Singapore, 841-872, 2000.
[3] http://en.wikipedia.org/wiki/Text_analytics
[4] Mrs. Sayantani Ghosh, Mr. Sudipta Roy, and Prof. Samir K. Bandyopadhyay. ―A tutorial review on Text Mining Algorithms, in International Journal of Advanced Research in Computer and Communication Engineering, Vol. 1, Issue 4, 2012.
[5] http://www.scism.lsbu.ac.uk/inmandw/ir/jaberwocky.htm
[6] Ian H. Witten, ―Text mining, University of Waikato, Hamilton, New Zealand
[7] Johannes C. Scholtes. ―Text-Mining: The next step in search technology‖, DESI-III Workshop Barcelona, 2009.
[8] Johannes C. ScholtesA. Voutilainen. ―A syntax-based part of speech analyser‖. In Proc. of the Seventh Conference of the European Chapter of the Association for Computation al Linguistics, pages 157–164, Dublin. Association for Computational Linguistics, 1995.
[9] Vishal Gupta, Gurpreet S. Lehal, 2009. ―A Survey of Text Mining Techniques and Applications‖ in Journal of Emerging Technologies in Web Intelligence, Vol. 1 No. 1.
[10] Shiqun Yin Yuhui Qiu1,Chengwen Zhong, 2007. Web Information Extraction and Classification Method .IEEE
[11] Umefjord G, Hamberg K, Malker H, Petersson G Fam Pract, 2006. The use of an Internet-based Ask the Doctor Service involving family physicians: evaluation by a web survey, 159- 66.
[12] Widman LE, Tong DA Arch Intern Med. 1997, Requests for medical advice from patients and families to health care providers who publish on the World Wide Web. 209-12.
[13] Text Mining Summit Conference Brochure, http://www.textminingnews.com/, 2005