A PROJECT REPORT ON

"SENTIMENT ANALYSIS OF EMAILS"

BY

Rucha Kshirsagar	800956296
Aasrita Kadiyala	800963238
Vinusha Bobburu	800969202
Neha Dalvi	800970459
Deepti Joshi	800960753

Under the Guidance of
Dr. Ali Sever

DEPARTMENT OF COMPUTER SCIENCE

COLLEGE OF COMPUTING AND INFORMATICS

CHARLOTTE, NORTH CAROLINA-28223

SPRING 2017

ABSTRACT

Sentiment analysis is that branch of Natural Language Processing which must handle multiple levels of granularity and it involves the generation and use of word lexicons. A large part of a person’s digital past is often stored in textual form, such as email messages or tweets or status updates or SMS texts and blog posts. One can perceive these text archives to be a personal informatics system in order to capture deep and useful information for the user that has some meaning. In general terms, sentiment analysis determines the overall tone or contextual polarity of the text with respect to a writer’s topic. However, it becomes a significant challenge for browsing and extracting useful information from an unstructured text corpus prevailing from many years.

In this project, we tend to use the techniques of sentiment analysis on personal text archives of employees of an organization in order to aid the management in the task of personal reflection and analysis which further helps them in taking appropriate decisions for the overall improvement of a company’s working environment in the near future. Because of this analysis, the company’s human resources department can determine the employee satisfaction by approximating the positivity or negativity of the entire organization’s communication through the help of emails which are both sent and received by the employees.

CONTENTS

CHAPTER 1

INTRODUCTION

Emotions are considered as an integral part of human perception and communication with the world. These emotions can be conveyed through our gestures, facial expressions, speech, and through our writing. For instance, any given sentence may be appropriate to many individual entities and the challenge lies in resolving the emotions those are evoked by one entity in another —where in requiring information that is not present in the sentence itself. Different persons express or feel certain emotions and this identification of emotions related to certain entities requires the analysis of the context, entity behavior and world knowledge in addition to the target sentence.

Email, which is a tool that is invented over 45 years ago, remains the most trusted form of online interaction till today for the reason it stands decentralized in a world of social applications. People frequently send emotional messages of love, joy, and condolence through emails. Hence sentiment analysis of personal email archives presents a good direction to pursue. The benefits of automatic analysis and tracking of emotions in personal mails include:

1. Decision Support Tool: It helps physicians to identify patients with a higher likelihood of attempting suicide.

2. Social Analysis: One can understand how genders usually communicate through work-place and personal email.

3. Productivity and Self-Assessment Tool: Helps to track emotions towards people and entities, over time.

4.Health Applications: Determines whether there is any chance of correlation among the emotional contents of letters and changes in a person's physiological, social or economic state.

5. Search: Enables affect-based search. Like in case of efforts to improve customer satisfaction may be aided by searching the received mail for snippets that express anger.

6. Writing Aids: It assists people in writing emails which convey the desired emotion, probably avoiding misinterpretation.

The goal of the project is to examine sentiments of emails stored in email server using R language. The project will analyze the sentiments of emails sent and received in an organization. The emails are provided in a CSV file. The sentiment analysis is done using R language by some predefined or some user defined functions in R.

CHAPTER 2
TECHNICAL DOCUMENTATION

2.1 PROGRAMMING LANGUAGE (R):

The aim of our project is to analyze the sentiments of emails and to display the statistical representation of the results in graphical charts and wordcloud. The R language provides a very good and sophisticated programming environment for statistical computing, visualization and data science and for data analytics. We chose R primarily although we have many options like SAS, because it is free, open-source and has a healthy library support and easy to implement.

2.2 LIBRARIES AND SOFTWARE PACKAGES USED:

One of the benefit of using R language is that there are various libraries available which implement a wide variety of statistical and graphical techniques, including linear and nonlinear modelling techniques, classical statistical tests, time-series analysis, classifications, clustering or grouping, and others which are easy to implement. R language is extendible through functions and extensions. R is highly usable though the use of user-submitted packages for specific functions are used for specific areas of study.

Following are the libraries that we used for our project.

A. plyr: This library contains a set of tools that solves a common set of problems: we need to slice a whole problem down into manageable small pieces, operate on each slice and then put all the slices or pieces together.

B. stringr: stringr is a set of simple wrappers that make R's string functions more consistent, and easier to use. It does this by ensuring that: function and argument names are consistent, all functions deal with NA's and zero length character appropriately, and the output data structures from each function matches the input data structures of other functions.

C. e1071: The e1071 library has functions which are useful for latent class analysis, fuzzy clustering, support vector machines, bagged clustering, short time Fourier transform and shortest path computation.

D. data.table: For a faster development, this library offers a flexible and natural syntax. The features of this library are fast aggregation of large data, fast add/modify/delete of columns, fast ordered joins, list columns and a fast file reader (fread).

E. rmongodb: This is an R extension supporting access to MongoDB.

F. tm: It contains a framework for text mining applications within R.

G. wordcloud: Word clouds are created using this library.

H. MASS: 'Modern Applied Statistics with S' contains functions and data sets that supports Venables and Ripley,

I. RODBC: Package RODBC implements ODBC database connectivity.

J. SnowballC: This dynamically determines the names of the languages for which stemming is currently supported by this package.

K. plotrix: This library is used to plot the data, various labeling, axis and color scaling functions used to create graphs like pie chart.

AFINN:

For the purpose of finding out the connotation and charge of a word, we used AFINN word list. AFINN provides a list of English words rated for valence with an integer between minus five and plus five. Minus five being a word with most negative connotation/charge while plus five being the one with most positive connotation/charge.

2.3 TOOLS:

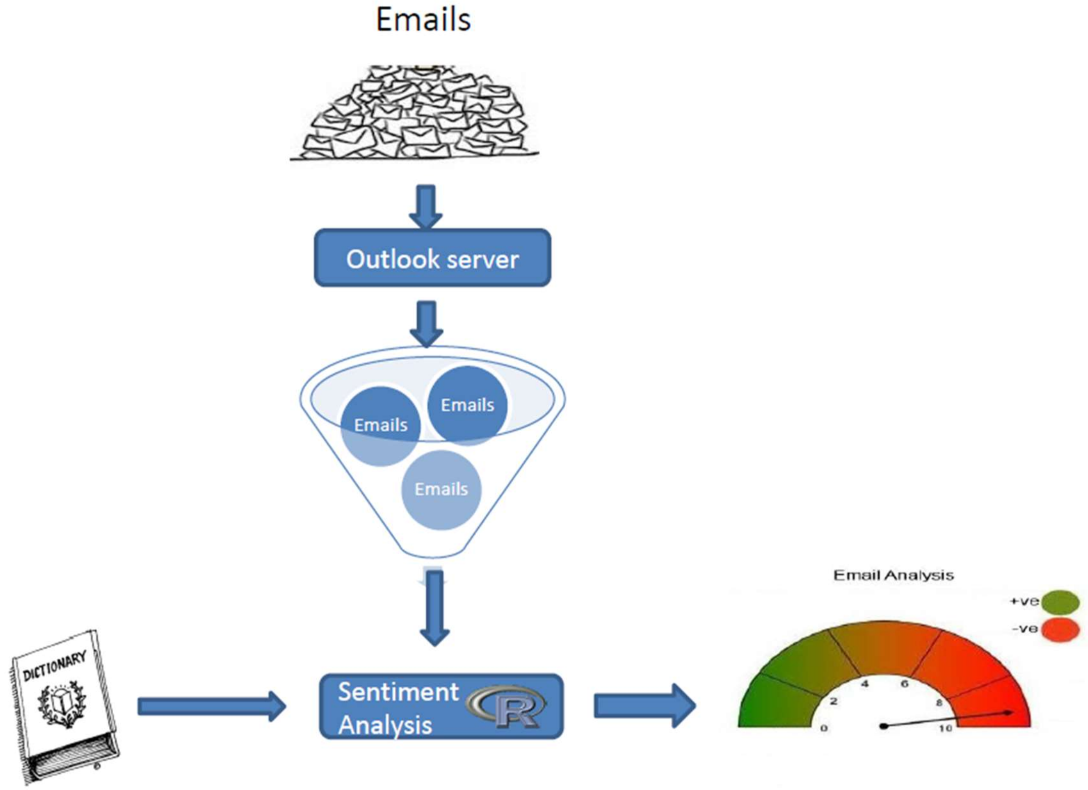
R: R is a language and environment for statistical computing and for visualizing graphs of data. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lu cent Technologies) by John Chambers and colleagues.

RStudio: RStudio is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management. RStudio is available in open source and commercial editions and runs on the desktop (Windows, Mac, and Linux) or in a browser.

R Studio can be used instead of development environment. Compare to other tools, R Studio has better debugging tools as well as the graphs produced by RStudio are visually better compared to R environment. Graphs produced by Rstudio are easy to understand for the users.

CHAPTER 3
SYSTEM ARCHITECTURE

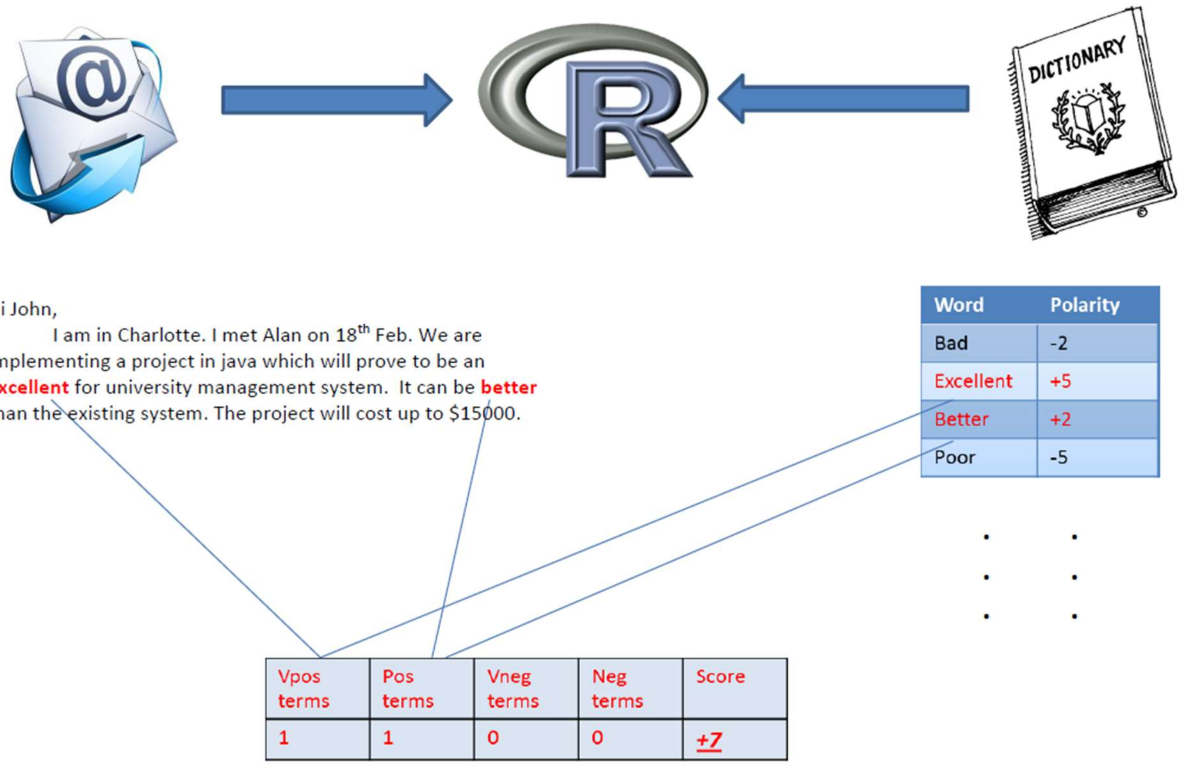
This is the system architecture of our project. Emails are extracted from the outlook server of an organization. We have used a predefined word list which has associated score for every word. The text in the emails is matched with the words from the dictionary and based on the resultant score of the emails the sentiment analysis of the emails is performed. The result is displayed in the form of charts and word cloud.



CHAPTER 4

ALGORITHM

The below diagram depicts the typical scenario in the project. R matches the words in the e mail with those in dictionary. The dictionary has words divided in four categories, positive, very positive, negative, very negative. Each category has a score associated with it. Total score is calculated by addition of scores of individual emails.



Hi John,

I am in Charlotte. I met Alan on 18th Feb. We are implementing a project in java which will prove to be an **excellent** for university management system. It can be **better** than the existing system. The project will cost up to $15000.

Word	Polarity
Bad	-2
Excellent	+5
Better	+2
Poor	-5
.	.
.	.
.	.

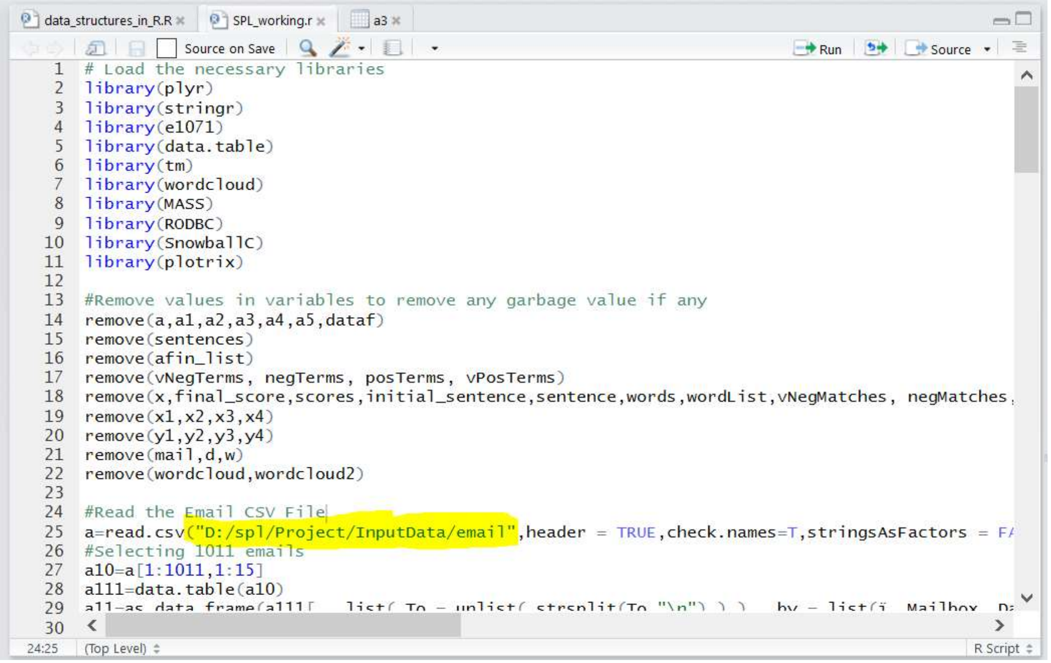
Vpos terms	Pos terms	Vneg terms	Neg terms	Score
1	1	0	0	+7

CHAPTER 5

PROJECT EXECUTION

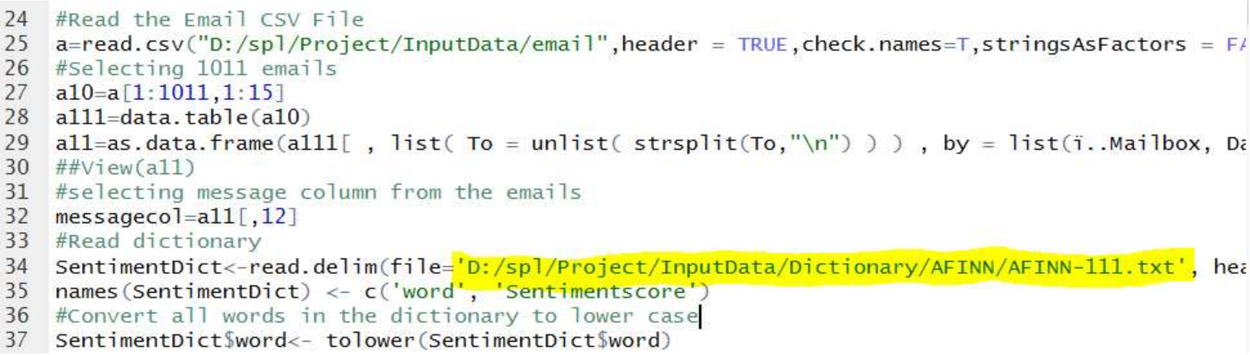
Following are the steps to run the project:

1. Start RStudio and open SPL.R script.
2. Update the location of the email input file (email) in SPL.R, as shown in the below diagram.



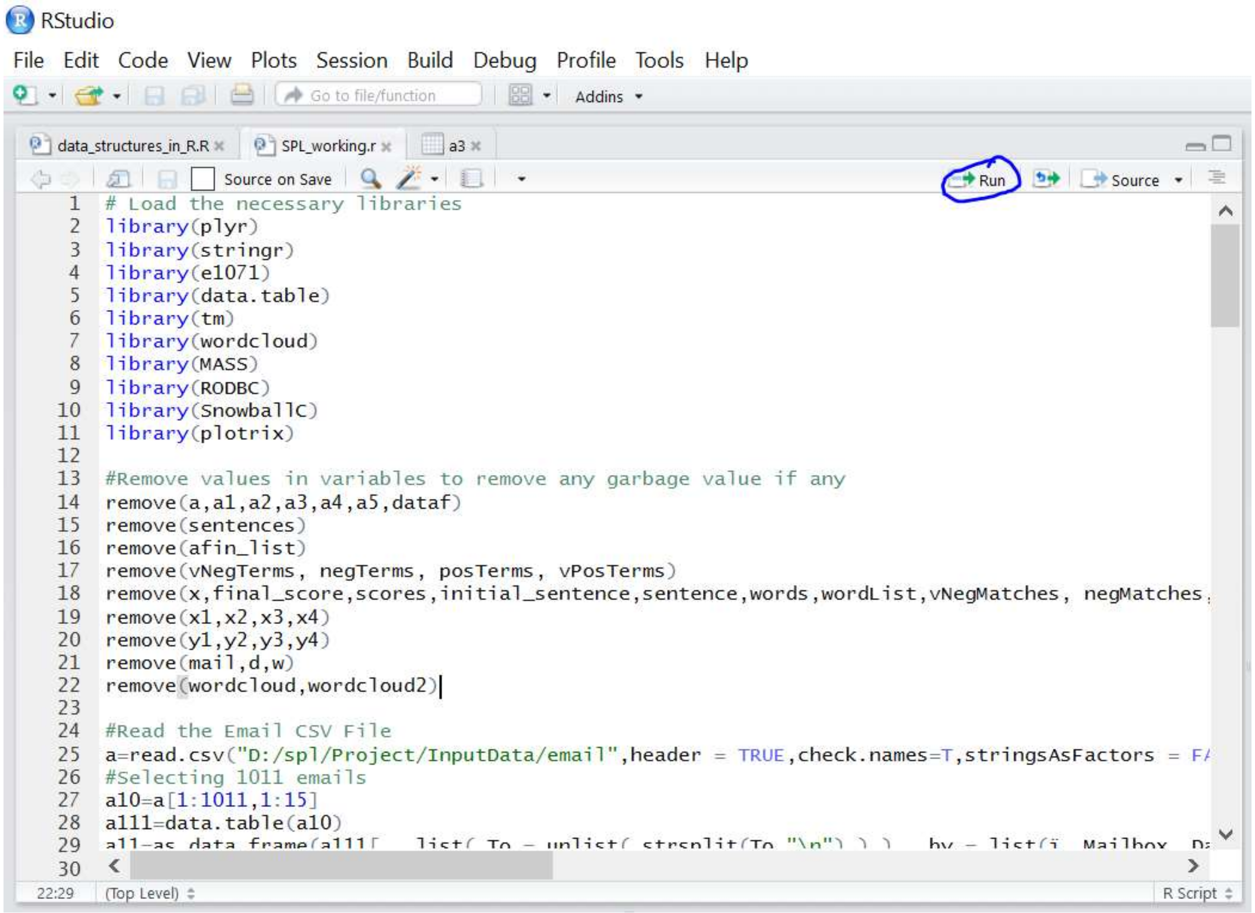
```
# Load the necessary libraries
library(plyr)
library(stringr)
library(e1071)
library(data.table)
library(tm)
library(wordcloud)
library(MASS)
library(RODBC)
library(SnowballC)
library(plotrix)

#Remove values in variables to remove any garbage value if any
remove(a,a1,a2,a3,a4,a5,dataf)
remove(sentences)
remove(afin_list)
remove(vNegTerms, negTerms, posTerms, vPosTerms)
remove(x,final_score,scores,initial_sentence,sentence,words,wordList,vNegMatches, negMatches,
remove(x1,x2,x3,x4)
remove(y1,y2,y3,y4)
remove(mail,d,w)
remove(wordcloud,wordcloud2)

#Read the Email CSV File
a=read.csv("D:/spl/Project/InputData/email",header = TRUE,check.names=T,stringsAsFactors = FA
#Selecting 1011 emails
a10=a[1:1011,1:15]
a111=data.table(a10)
a11=as.data.frame(a111[ , list( To = unlist( strsplit(To,"\n") ) ) , by = list(ï..Mailbox, Da
```

3. Update the location of the AFINN dictionary in SPL.R, as shown in the below diagram.



```
#Read the Email CSV File
a=read.csv("D:/spl/Project/InputData/email",header = TRUE,check.names=T,stringsAsFactors = FA
#Selecting 1011 emails
a10=a[1:1011,1:15]
a111=data.table(a10)
a11=as.data.frame(a111[ , list( To = unlist( strsplit(To,"\n") ) ) , by = list(ï..Mailbox, Da
##View(a11)
#selecting message column from the emails
messagecol=a11[,12]
#Read dictionary
SentimentDict<-read.delim(file='D:/spl/Project/InputData/Dictionary/AFINN/AFINN-111.txt', hea
names(SentimentDict) <- c('word', 'Sentimentscore')
#Convert all words in the dictionary to lower case
SentimentDict$word<- tolower(SentimentDict$word)
```

4. Click on ‘Run’ button as shown in the below diagram.



```r
# Load the necessary libraries
library(plyr)
library(stringr)
library(e1071)
library(data.table)
library(tm)
library(wordcloud)
library(MASS)
library(RODBC)
library(SnowballC)
library(plotrix)

#Remove values in variables to remove any garbage value if any
remove(a,a1,a2,a3,a4,a5,dataf)
remove(sentences)
remove(afin_list)
remove(vNegTerms, negTerms, posTerms, vPosTerms)
remove(x,final_score,scores,initial_sentence,sentence,words,wordList,vNegMatches, negMatches,
remove(x1,x2,x3,x4)
remove(y1,y2,y3,y4)
remove(mail,d,w)
remove(wordcloud,wordcloud2)

#Read the Email CSV File
a=read.csv("D:/spl/Project/InputData/email",header = TRUE,check.names=T,stringsAsFactors = FA
#Selecting 1011 emails
a10=a[1:1011,1:15]
a111=data.table(a10)
a11=as.data.frame(a111[ , list( To = unlist( strsplit(To,"\n") ) ) , by = list(i, Mailbox, Da
```

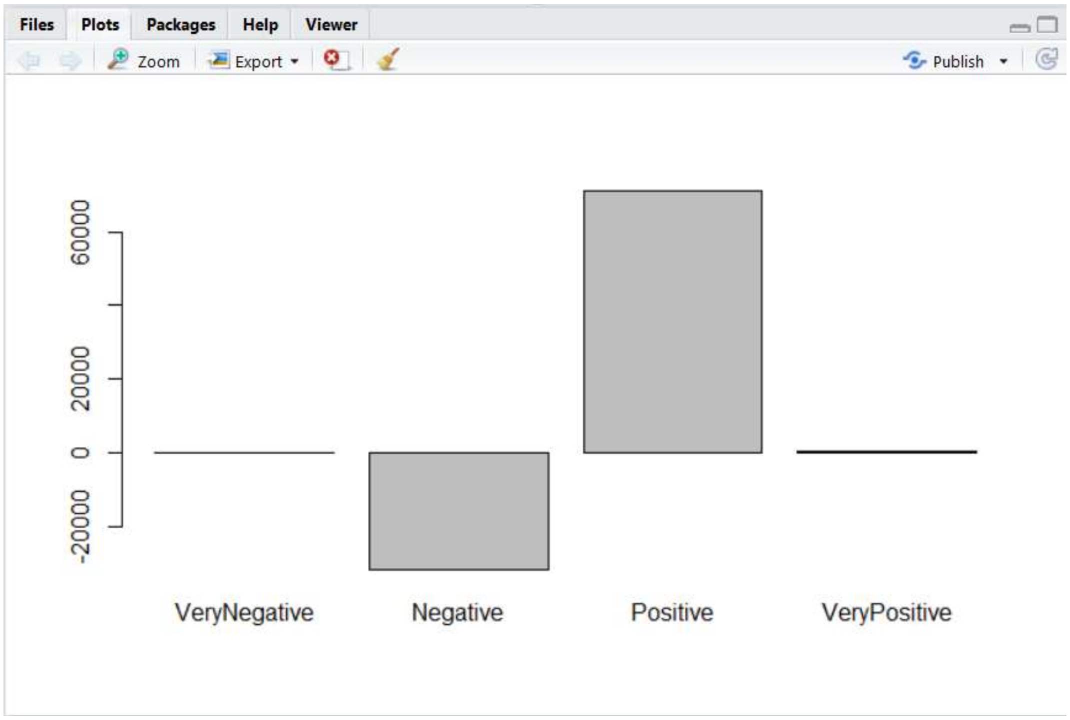
CHAPTER 6

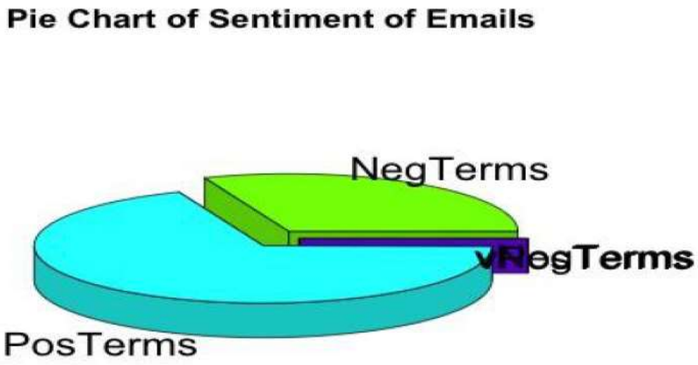
RESULT

6.1 INPUT AND OUTPUT:

Once we execute the script, output is shown in the form of Bar Chart as well as Pie Chart. By analyzing both of this chart we can conclude that the overall sentiment of a given organization is positive. we can generate word cloud also.

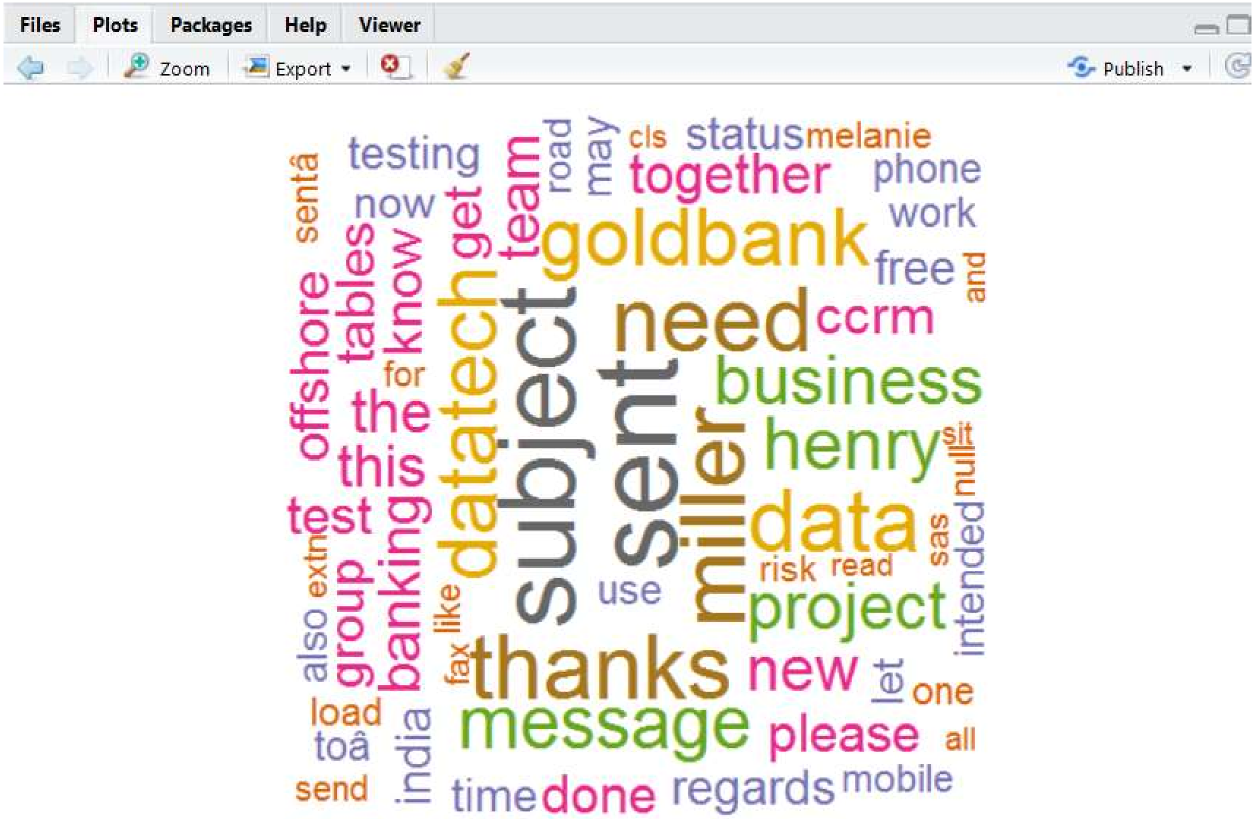
Bar Chart of Sentiments of Emails





The following graph shows which most frequently used words in the emails.

Word Cloud



CHAPTER 7
CONCLUSION AND FUTURE SCOPE

We have learned how R programming language can be successfully used in statistical and data analysis especially, to analyze the sentiments of emails. All the outstanding requirements have been successfully satisfied by the software. The software has simplified the tedious work of analysis of emails and provided the output in the form of bar as well as pie chart which are easy to comprehend. The email analysis provided by the software can potentially play an important role in helping the management as well as human resources department in judging the positivity and negativity in employee communication.

As of now, we have not considered if there is some content in the email which sarcastic. However, we can implement an algorithm to determine this. This will be a good add-on to our project.

This project can be easily scaled in future to analyse online product review content. For example, consider a user wants to buy a new smart-phone. Currently, he needs to view customer reviews for that product separately on different websites, like Amazon.com, etc. We can extend our system to analyse the reviews from different websites and show the analysis at a single place in graphical charts which can save a lot of time of the user.

CHAPTER 8

REFERENCES

1. http://www.r-project.org/about.html
2. http://en.wikipedia.org/wiki/R_(programming_language)
3. http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010
4. http://andybromberg.com/sentiment-analysis/
5. http://www.ellicium.com/
6. http://www.rstudio.com/products/rstudio/
7. https://mobisocial.stanford.edu/papers/informatics11.pdf
8. https://pdfs.semanticscholar.org/7207/8f42884165f073c23577a555e0c21ad5fa74.pdf