Assignment 8

CS 553

Twitter Sentiment Analysis

Project Report

Group 12

Manasa Geemarahalli Prakash 10457598

Mahvish Syed 10456845

Shweta Singh 10457493

Glen D’mello 10452455

* **Introduction:**

The Twitter messages created via Twitter are named as Tweets. These data are available in public domain. It can thus be taken as raw data primarily for the extraction of opinions, for the analysis of customer fulfillment and for different rating policy schemes and, ultimately, a study of sentiment has been conducted. This Project study has been conducted to identify the sentiments of the people of India infected with COVID19 and identifies what emotions people have been sharing from different parts of the India.

The tweets posted in English have been considered for a sentiment analysis to understand how people from different infected countries have responded during this pandemic situation to cope with it. The collected tweets will be used, preprocessed and applied with text mining algorithms for performing the sentiment analysis.

This project aims to capture, process and evaluate people's feelings within the certain timeframe on the tweets posted on twitter. The study would therefore concentrate on the following questions:

1. **Collect** the tweets through Twitter API using RTweet package in R programming was used. The Hashtag used for collecting the tweet is #covidIndia
2. **Preprocess** the tweets by data cleaning (removing white spaces, links, punctuations, stop words, tokenization, retweet).
3. **Calculate** the sentiment using syuzhet package and analyze the result.

* **Steps to install R studio and execute code:**

1. Install R studio using below link for Macbook ,Ubuntu or windows

<https://www.datacamp.com/community/tutorials/installing-R-windows-mac-ubuntu>

1. To run code in R, Select code that you want to execute and click on Run button.
2. Output will be display at bottom screen and plots/variables are displayed on side panel.

* **Setup R enviornment and Library installation**

Remove any initialize variable and clean environment using below code

rm(list= ls())

Installation of the needed packages

list.of.packages <- c("twitteR", "tm", "SnowballC", "ggplot2", "RColorBrewer", "wordcloud", "topicmodels", "data.table" ,"syuzhet","lubridate","scales","reshape2","qplyr")

new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]

if(length(new.packages)) install.packages(new.packages)

install.packages("qplyr")

install.packages("rtweet")

install.packages("twitteR")

library("twitteR")

library(rtweet)

library(dplyr)

## ***twitteR***

twitteR is an R package which provides access to the Twitter API. Most functionality of the API is supported, with a bias towards API calls that are more useful in data analysis as opposed to daily interaction.

## ***rtweet***

R client for accessing Twitter’s REST and stream APIs.

**Note : Make sure to restart your R session after installing the packages.**

* **Twitter API setup:**

The below code basically extracts tweets from Twitter using the Twitter API. For extracting the tweets we need to provide some Access Token which I have provided in the setup\_twitter\_oauth() function.

Setup API using oAuth API key.

setup\_twitter\_oauth(

'pq9cNg3BUU2EqIUDbi75oRRdq', 'P88EPX1l1aMnoDnFtdsO3iOZcFveKM5vIuLUNuo3W1aDzichFd',

'118425371-Bnx7lyzFWytRzORMCKtgHMTx226wpbtgC7QqpcOF',

'f07cgIaaCkpxT4B64EJBqQ560iNIZiGRf1kisb3mClceW')

## ***Setup\_twitter\_oauth***

This function wraps the OAuth authentication handshake functions from the httr package for a twitteR session

Arguments:

consumer\_key: Sets up the OAuth credentials for a twitteR session.

consumer\_secret: The consumer key supplied by Twitter

access\_token: The consumer secret supplied by Twitter

access\_secret: The access token supplied by Twitter

* **Extraction Of the Data:**

We have extracted 4000 tweets which is English language for #covidIndia.

For the tweet collection, RTweet package in R programming was used.

The Hashtag used for collecting the tweet were #covidIndia and the collected Tweets saved in CSV file.

The tweets were filtered out while collecting the tweets to avoid duplication of the tweets. As the complete database was obtained, the data cleaning process has been performed, where the white spaces, punctuation, stop words were removed.

1. **Code for data extraction:**

covidTweets <- search\_tweets("#covidIndia", n = 4000, include\_rts = FALSE, lang="en")

n.tweet <- length(covidTweets)

tweets.df <- as.data.frame(covidTweets) #converting the lists of tweets into dataframe

View(tweets.df)

new\_df <- tweets.df[,5] ## Selecting only the 5th column from the data frame

print(new\_df)

***Search\_tweets***

Returns Twitter statuses matching a user provided search query. ONLY RETURNS DATA FROM THE PAST 6-9 DAYS. To return more than 18,000 statuses in a single call, set "retryonratelimit" to TRUE.

search\_tweets2 Passes all arguments to search\_tweets. Returns data from one OR MORE search queries.

1. **Save data in csv file for offline use (File attached: covid.csv)**

write.csv(new\_df, file = "/Users/mahvishsyed/Desktop/Covid.csv")

1. **Transformation**

By default, the tweets are in **list** format but for analysis purpose we need to first convert it into **dataframe** and then select only the column that contains text data for creating the text corpus.

library(tm)

print(new\_df)

# build a corpus, and specify the source to be character vectors

myCorpus <-Corpus(VectorSource(iconv(new\_df, "utf-8", "ASCII", sub="")))

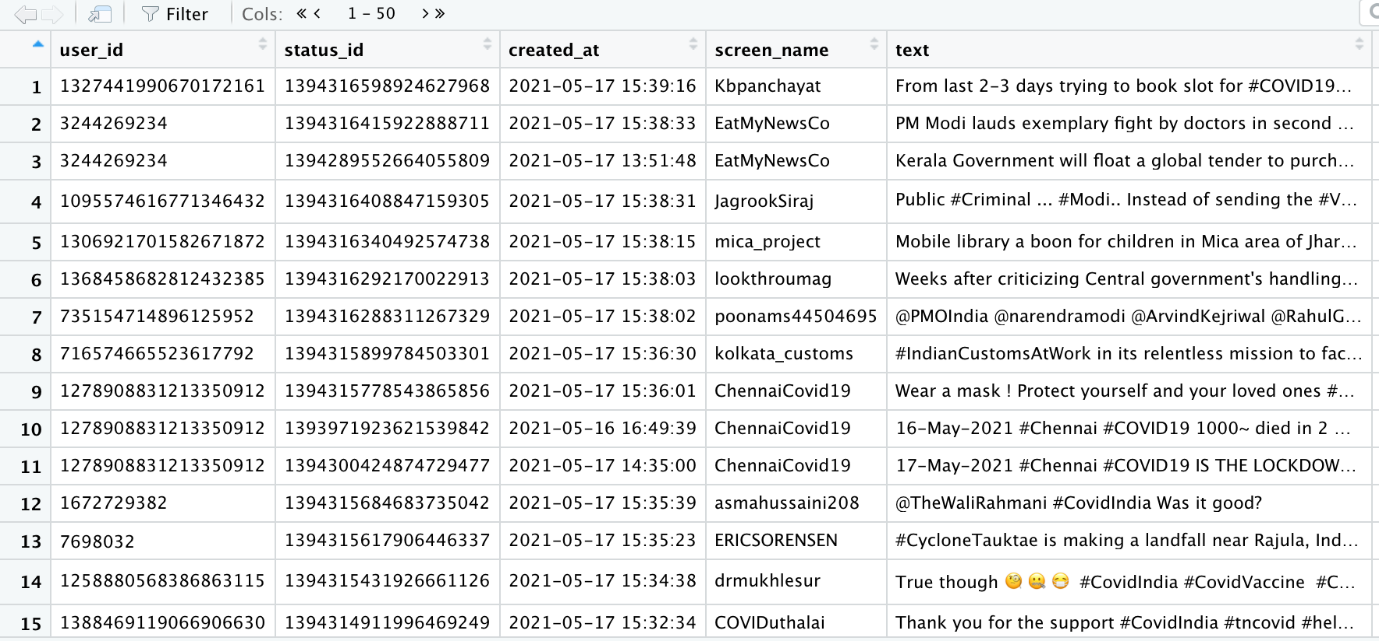
x<-as.character(myCorpus) # We will need this for sentiment analysis so just copying the corpus to a varaible x

inspect(myCorpus[1:6]) # inspect the 1st 6 lines form the corpus

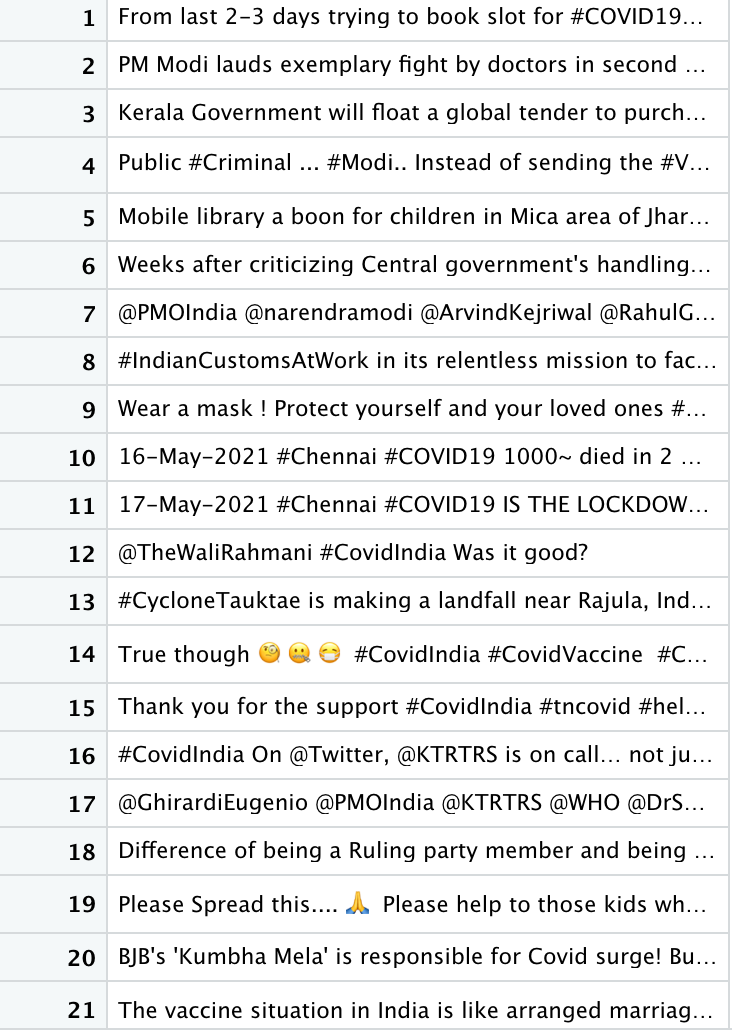
# convert to lower case

myCorpus <- tm\_map(myCorpus, content\_transformer(tolower))

1. **Snippet of data in DB form**



1. **Snippet of selected data:**



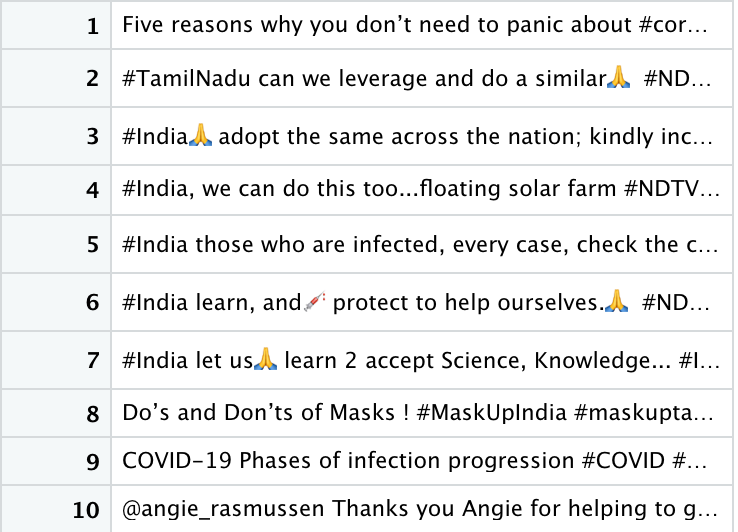
1. **Cleaning of the Data- Removing stop words, urls , numeric data and other stuff**

In the cleaning process will be performing all the data cleaning process, (since the data is textual data) the cleaning process involves the following:

* + Converting the data to lower case
  + Removing the urls
  + Removing the punctuation marks
  + Removing Stop words
  + Removing English language space
  + Removing extra white space

**Note:- We have also removed Covid, India and CovidIndia from the corpus because it was obvious that the term will occur in almost all the tweets and it might hamper the prediction or could be too over fitted.**

* **Snippet of uncleaned data: (shown below first 20 data)**



* **Code to clean data**

library(tm)

View(new\_df)

# build a corpus, and specify the source to be character vectors

myCorpus <-Corpus(VectorSource(iconv(new\_df, "utf-8", "ASCII", sub="")))

x<-as.character(myCorpus) # We will need this for sentiment analysis so just copying the corpus to a varaible x

inspect(myCorpus[1:6]) # inspect the 1st 6 lines form the corpus

# convert to lower case

myCorpus <- tm\_map(myCorpus, content\_transformer(tolower))

# remove URLs

removeURL <- function(x) gsub("http[^[:space:]]\*","", x)

myCorpus <-tm\_map(myCorpus, content\_transformer(removeURL))

#inspect(myCorpus[1:6])

# remove anything other than English letters or space

removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]\*","",x)

myCorpus <- tm\_map(myCorpus, content\_transformer(removeNumPunct))

#inspect(myCorpus[1:6])

# Removing extra add words which I feel are unnessary

myStopwords <- c(stopwords('english'),"This","a","of","the","was","is", "that", "covid", "covidindia", "india")

myCorpus <- tm\_map(myCorpus, removeWords, myStopwords)

#remove extra whitespace

myCorpus <- tm\_map(myCorpus, stripWhitespace)

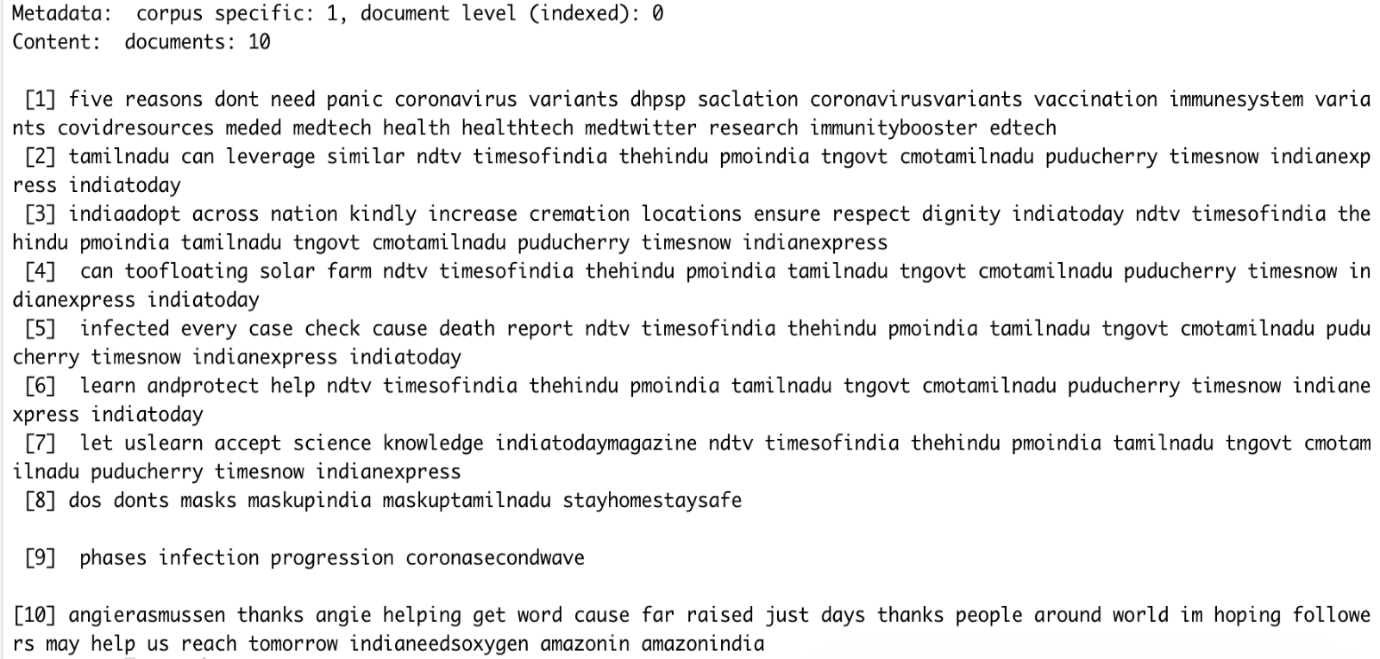
# keep a copy of corpus to use later as a dictionary for stem completion

myCorpus <- tm\_map(myCorpus, removeNumbers)

#inspect(myCorpus[1:10])

myCorpusCopy <- myCorpus

* **Snippet of Clean data : (first 20 data )**



* **Loading Data**

## Created Term-Document Matrix that gives information about terms considered from documents.

The following snippet shows the numers of terms = **6667** in documents= **1269** also it shows the Non-/sparse enteries **18832/8441591** along with the weighing term used i.e **term freq** and other info like sparsity, max length along with the sample of 10 terms. Along with the weighing term used i.e term freq and other info like amp, cases, covidemergency, covidhelp, help, oxygen , people,please, verified, help.

* **Code for creating Term Document Matrix**

Constructs or coerces to a term-document matrix or a document-term matrix.

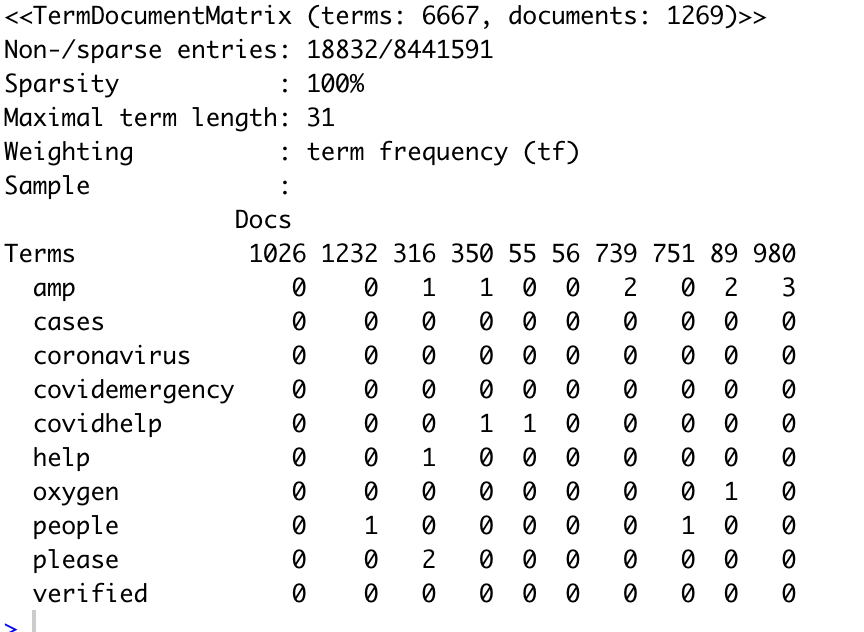
tdm <- TermDocumentMatrix(myCorpus, control = list(wordLengths = c(1,Inf)))

?TermDocumentMatrix

tdm

inspect(tdm)

**Snippet of Term-Documented matrix**



*Fig : Term Document Matrix*

* **Frequency Plotting**

The frequency plot basically plots the term with frequency more than 100 i.e the terms occurring in atleast 100 document. We have kept the lower threshold so as to get significant terms. The most frequent terms used are **covidindia** followed by **covid**.

#inspect frequent words

(freq.terms <- findFreqTerms(tdm, lowfreq = 100)) # finding word freq here i am only considering words with occcur in atleast 100 docs to

# avoid over crowding

# Creating a named vector with the frequency of the words

term.freq <- rowSums(as.matrix(tdm))

term.freq <- subset(term.freq, term.freq >= 100)

# Transforming a named vector to a dataframe

df <- data.frame(term = names(term.freq), freq = term.freq)

library(ggplot2)

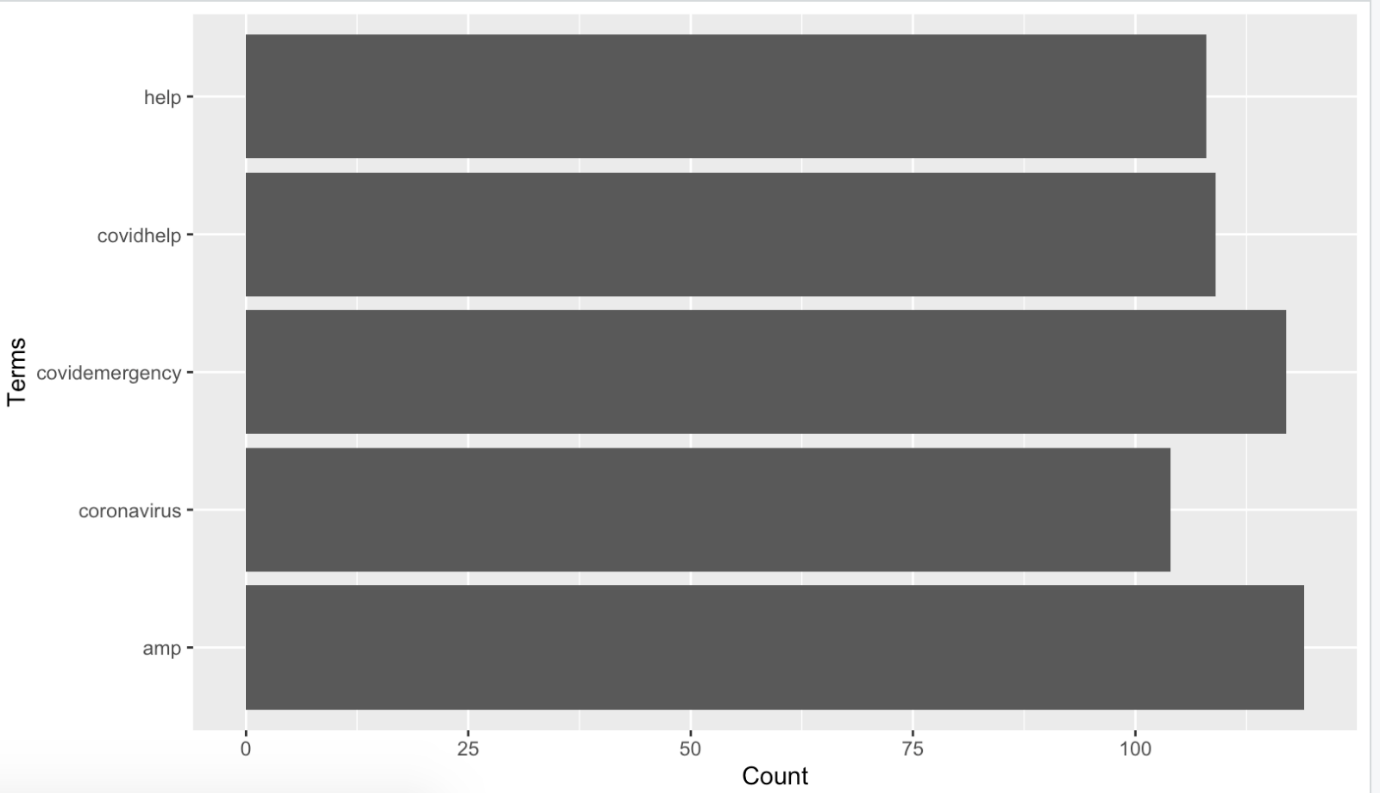
p <- ggplot(df, aes(x = term, y = freq))

p <- p + geom\_bar(stat = "identity")

p <- p + xlab("Terms") + ylab("Count")

p <- p + coord\_flip()

print(p)



*Fig: Frequency Plot*

Observation:

* From graph you can see, “covid emergency” is high. Taking into consideration of current situation in India, covid in India has reached to threshold.
* Emergency cases are really high and even facilities are needed with high priority.
* Also “help” word is second highly frequency. With limited resource, India seeking for help from other countries and also within countries.
* People helping each other to provide essentials facilities.
* **Word Association and Corelation**

(freq.terms <- findFreqTerms(tdm, lowfreq = 50))

Word Association is used to see how closely two terms are associated with each other. It also plays a significant role in text analysis because it provides us with a list of closely associated words and by learning the association we can learn about the realtion between two terms. For our project we found some associations between covid, vaccine and government.

1. Finding some associations between covid, vaccine and government

findAssocs(tdm, "covid", 0.1)

# findAssocs is used to find the association of various words with

#respect to a particular word provided in " ".

findAssocs(tdm, "vaccine", 0.2)

findAssocs(tdm, "government", 0.1)

**Output Snippet:**

Table

Description automatically generated

*Fig : Association between terms*

A picture containing text, receipt

Description automatically generated

*Fig: Association between terms*

A picture containing text, receipt

Description automatically generated

*Fig: Association between terms*

Above are the three outputs that depicts that the term covid, vaccine and government is associated with terms like vacciated, appointment and it makes sense because of recent situations in India.

**Observation:**

The word association shows following result:

1- For “covid” word, can see that its information are available at social platform like facebook and youtube.Hence relation is shown for them.

Also people are treated but many scumb

* **Sentiment Analysis**

It is the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc., is positive, negative, or neutral.

For prediting the sentiment analysis we are using R’s syuzhet, lubridate, ggplot2 libraries. R learns from the terms in our corpus and classifies it into 7-10 categories namely, positive, negative, fear, anger, trust, surprise, sadness, joy, disgust.

library(syuzhet), library(lubridate) ,library( ggplot2), library(reshape2), library(qplyr)

sentiment <- get\_nrc\_sentiment(x)

t<-as.matrix(sentiment)

#write.csv(sentiment, ""/Users/mahvishsyed/Desktop/CS-553/Text Mining Project/MySentiments.csv")

getwd()

comments <- cbind(new\_df,sentiment)

sentimentTotals <- data.frame(colSums(sentiment[,c(1:8)]))

names(sentimentTotals) <- "count"

sentimentTotals <- cbind("sentiment" = rownames(sentimentTotals), sentimentTotals)

rownames(sentimentTotals) <- NULL

ggplot(data = sentimentTotals, aes(x = sentiment, y = count)) +

geom\_bar(aes(fill = sentiment), stat = "identity") +

theme(legend.position = "none") +

xlab("Sentiment") + ylab("Total Count") + ggtitle("Total Sentiment Score for all Tweets")

Chart, bar chart

Description automatically generated

*Fig: Sentiment Analysis*

**Observation:**

* As we can see in the graph above, the opinions of tweets have fear, sadness among but it is strange to see that there is also a high percent of trust factor involved in the peoples sentiments .
* **Hierarchial clustering**

In [data mining](https://en.wikipedia.org/wiki/Data_mining) and [statistics](https://en.wikipedia.org/wiki/Statistics), hierarchical clustering (also called hierarchical cluster analysis or HCA) is a method of [cluster analysis](https://en.wikipedia.org/wiki/Cluster_analysis) which seeks to build a [hierarchy](https://en.wikipedia.org/wiki/Hierarchy) of clusters. We have used divisive clusterig algorithm which a top-dow appraoch usig dendograms.

# remove sparse terms

tdm2 <- removeSparseTerms(tdm, sparse = 0.9)

# Showing the terms that are left for the analysis

print(dimnames(tdm2)$Terms)

m2 <- as.matrix(tdm2)

# cluster terms

distMatrix <-dist(scale(m2))

fit <- hclust(distMatrix, method = "ward.D2")

p <- plot(fit)

p <- rect.hclust(fit, k = 4) # fit into 6 clusters

print(p)

# Showing the groups

(groups <-cutree(fit, k = 4))

print(groups)

Diagram, schematic

Description automatically generated

*Fig: Hierarchical clustering*

**Observation:**

From the above snippit you can see how the terms hours, deaths, reports, active, covidindia are closely related to each other and thus clustered in kind of common cluster.

* **Topic Modelling**

In [machine learning](https://en.wikipedia.org/wiki/Machine_learning) and [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing), a topic model is a type of [statistical model](https://en.wikipedia.org/wiki/Statistical_model) for discovering the abstract "topics" that occur in a collection of documents. Topic modeling is a frequently used text-mining tool for discovery of hidden semantic structures in a text body.

For our project we have used LDA i.e Latent Dirichlet Association which basically takes the most frequent occurring terms from the documents and then creates topic models based on those terms.

dtm <- as.DocumentTermMatrix(tdm)

library(topicmodels)

library(data.table)

lda <- LDA(dtm, k = 8) # find 8 topics

(term <- terms(lda,6)) # first 6 terms of every topic

term <- apply(term, MARGIN = 2, paste, collapse = ", ")

# first topic identified for every document (tweet)

topic <- topics(lda, 1)

topics <- data.frame(date=as.IDate(tweets.df$created), topic)

p <- qplot(date, ..count.., data=topics, geom = "density", fill = term[topic], position="stack")

print(p)

Chart

Description automatically generated

*Fig 8: Topic Modeling*

**Observation:**

The plot shows that the LDA function has created 8 topic models.