Program – 10

AIM: Write a program to find similar documents with cosine similarity.

Introduction and Theory

Cosine Similarity

Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. The smaller the angle, higher the cosine similarity.

The cosine for two non-zero vector can be calculated from the Euclidean dot product.

$$A \cdot B = ||A|| \, ||B|| \cos(\theta)$$

The cosine similarity is then calculated as:

$$similarity = \cos(\theta) = \frac{A \cdot B}{||A|| \, ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Code

Cosine.R

```
library(twitteR)
library(tidytext)
library(dplyr)
library(tm)
library(SnowballC)

workingDir <- "~/Documents/R programs"
setwd(workingDir)

url <- "http://www.rdatamining.com/data/rdmTweets-201306.RData"
download.file(url, destfile = "rdmTweets-201306.RData")

load(file = "rdmTweets-201306.RData")
tweets <- twListToDF(tweets)
load(file = "rdmTweets-201306.RData")
tweets <- twListToDF(tweets)

tweets <- tweets %>%
  mutate(text=gsub("(http|https).+$|\\n|&amp|[[:punct:]]","",text),
         rowIndex=as.numeric(row.names(.))) %>%
  select(text,retweetCount,rowIndex)

docList <- as.list(tweets$text)
```

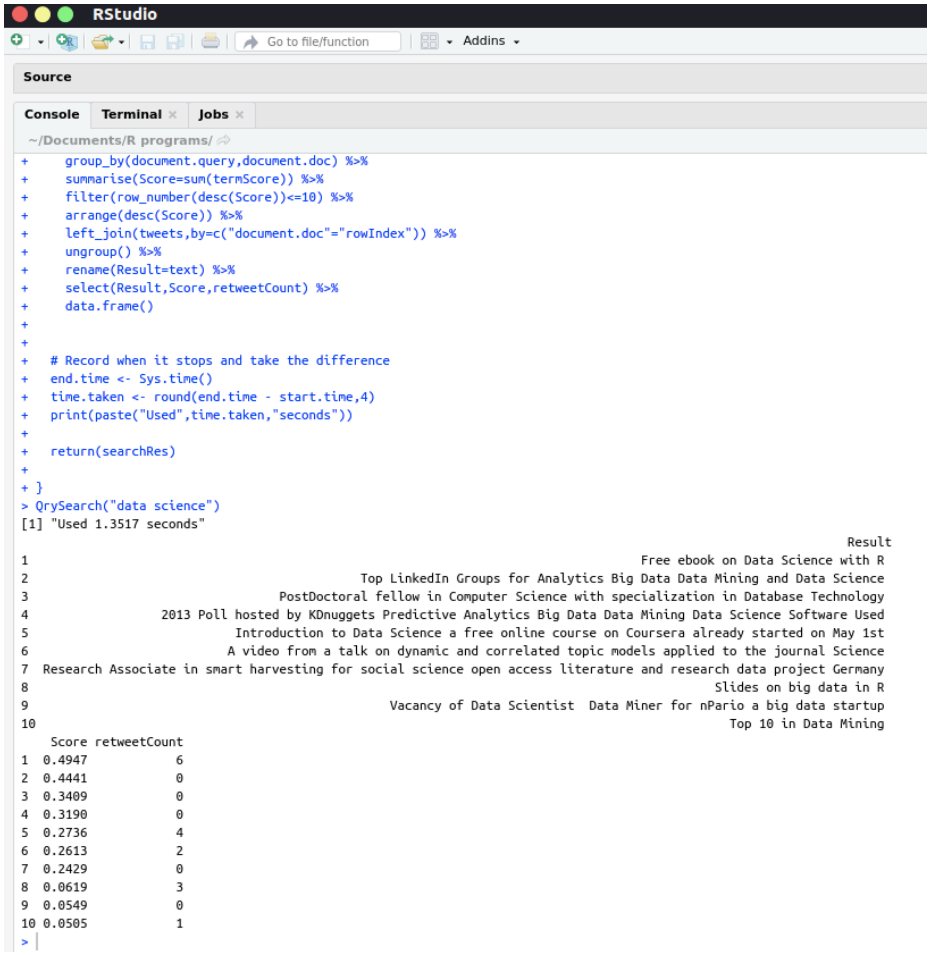
Program – 10

```
N.docs <- length(docList)

QrySearch <- function(queryTerm) {

  # Record starting time to measure your search engine performance
  start.time <- Sys.time()

  # store docs in Corpus class which is a fundamental data structure in text mining
  my.docs <- VectorSource(c(docList, queryTerm))


  # Transform/standaridze docs to get ready for analysis
  my.corpus <- VCorpus(my.docs) %>%
    tm_map(stemDocument) %>%
    tm_map(removeNumbers) %>%
    tm_map(content_transformer(tolower)) %>%
    tm_map(removeWords,stopwords("en")) %>%
    tm_map(stripWhitespace)


  # Store docs into a term document matrix where rows=terms and cols=docs
  # Normalize term counts by applying TDiDF weightings
  term.doc.matrix.stm <- TermDocumentMatrix(my.corpus,
                                            control=list(
                                              weighting=function(x) weightSMART(x,spec="ltc"),
                                              wordLengths=c(1,Inf)))



  # Transform term document matrix into a dataframe
  term.doc.matrix <- tidy(term.doc.matrix.stm) %>%
    group_by(document) %>%
    mutate(vtrLen=sqrt(sum(count^2))) %>%
    mutate(count=count/vtrLen) %>%
    ungroup() %>%
    select(term:count)
  docMatrix <- term.doc.matrix %>%
    mutate(document=as.numeric(document)) %>%
    filter(document<N.docs+1)
  qryMatrix <- term.doc.matrix %>%
    mutate(document=as.numeric(document)) %>%
    filter(document>=N.docs+1)

  # Calcualte top ten results by cosine similarity
  searchRes <- docMatrix %>%
    inner_join(qryMatrix,by=c("term"="term"),
               suffix=c(".doc",".query")) %>%
    mutate(termScore=round(count.doc*count.query,4)) %>%
    group_by(document.query,document.doc) %>%
```

Program – 10

```
    summarise(Score=sum(termScore)) %>%
    filter(row_number(desc(Score))<=10) %>%
    arrange(desc(Score)) %>%
    left_join(tweets,by=c("document.doc"="rowIndex")) %>%
    ungroup() %>%
    rename(Result=text) %>%
    select(Result,Score,retweetCount) %>%
    data.frame()


  # Record when it stops and take the difference
  end.time <- Sys.time()
  time.taken <- round(end.time - start.time,4)
  print(paste("Used",time.taken,"seconds"))

  return(searchRes)

}

QrySearch("data science")
```

Results & Outputs



output

Program – 10

Findings and Learnings:

1. Cosine similarity is one of the most commonly used similarity metrics in text processing
2. R provides easy to use tools for performing text analysis.
3. We have successfully implemented document retrieval using cosine similarity.