Project 1 - Text Mining of Five Files

Arpan Sapkota

2023-04-19

Project 1 - Text Mining of Five Files

You must search and download first five (5) free pdf files on the topic of your choice from Google Scholar (https://scholar.google.com/)

You must put these five (5) pdf files in a folder called "MDS503P1"

Use the "pdftools" package to read these five pdf files in R

Once you read the text of these pdf files in R then create a "corpus" with tm package

Perform pre-processing of the corpus followed by getting term-document matrix to show the most frequent terms, word clouds with and without color, network graph, and topic modeling with comments/interpretation for each step as done in the class today

1. Loading PDF files related to the Healthcare Data Analytics

```
#1. Use the "pdftools" package to read the five PDF files in R:
library(pdftools)
```

```
## Using poppler version 22.02.0
```

```
# Set the working directory to the folder where the PDF files are stored
setwd("/Users/arpan/Desktop/MDS/01 MDS I-I/MDS 503 - Statistical Computing with R/Lab/MDS503P1")

# Read the five PDF files and store them in a list
pdf_files <- list.files(pattern = "*.pdf")
pdf_text <- lapply(pdf_files, pdf_text)

pdf_files
```

```
## [1] "Big Data Analytics for Healthcare Industry.pdf"
## [2] "Business Intelligence Framework for Healthcare Analytics.pdf"
## [3] "Designing Healthcare Analytics Solutions.pdf"
## [4] "Examining the Diagnosis Treatment Cycle.pdf"
## [5] "Healthcare Analytics - A Comprehensive Review.pdf"
```

2. Creating Corpus

```r
#2. Create a "corpus" with the "tm" package:

library(tm)
```

```
## Loading required package: NLP
```

```r
# Create a corpus from the pdf_text list
corpus <- Corpus(VectorSource(unlist(pdf_text)))

str(corpus)
```

```
## Classes 'SimpleCorpus', 'Corpus'  hidden list of 3
##  $ content: chr [1:49] "BIG DATA MINING AND ANALYTICS\nI S S N 22 2 0 9 6 - 0 6 54 1 1 0 5 / 0 6 1 1
##  $ meta   :List of 1
##   ..$ language: chr "en"
##   ..- attr(*, "class")= chr "CorpusMeta"
##  $ dmeta  :'data.frame': 49 obs. of  0 variables
```

```r
# Inspect the corpus to ensure it has been created correctly
#inspect(corpus)
inspect(corpus[1:1])
```

```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 1
##
## [1] BIG DATA MINING AND ANALYTICS\nI S S N 22 2 0 9 6 - 0 6 54 1 1 0 5 / 0 6 1 1 p p 4 8- 5 7\nVolum
```

3. Pre-processing of the Corpus

```r
#3. Perform pre-processing of the corpus:

# Convert all text to lowercase
corpus <- tm_map(corpus, content_transformer(tolower))

# Remove numbers, punctuation, and whitespace
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, stripWhitespace)

# Remove stop words
corpus <- tm_map(corpus, removeWords, stopwords("english"))

#removing unwanted html links and "\n" new line in the Corpus
corpus <- tm_map(corpus, content_transformer(function(x) gsub("http[^[:space:]]*","", x)))
corpus <- tm_map(corpus, content_transformer(function(x) gsub("\\n*","", x)))
corpus <- tm_map(corpus, content_transformer(function(x) gsub("-","", x)))
#inspect(corpus) #inspect the results in corpus

# Inspect the corpus to ensure pre-processing has been done correctly
inspect(corpus[1:1])
```

```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 1
##
## [1] big data mining  analytics  s s n l l l l p p  volume number march doi bdma big data analytics  l
```

4. Term-Document Matrix

```
#4. Get term-document matrix to show the most frequent terms:

# Create the term-document matrix
tdm <- TermDocumentMatrix(corpus, control = list(wordLength=c(1,Inf)))

# Get the most frequent terms cosidering lower frequency = 20
freq_terms <- findFreqTerms(tdm, lowfreq = 20)

# Inspect the most frequent terms
freq_terms
```

```
##   [1] "activities"    "also"          "analytics"     "applications"
##   [5] "based"         "better"        "big"           "care"
##   [9] "clinical"      "computer"      "data"          "decisions"
##  [13] "different"     "hadoop"        "health"        "healthcare"
##  [17] "improve"       "including"     "informatics"   "information"
##  [21] "knowledge"     "many"          "mining"        "need"
##  [25] "new"           "number"        "patient"       "prediction"
##  [29] "provide"       "providers"     "quality"       "records"
##  [33] "research"      "results"       "science"       "sector"
##  [37] "sources"       "storage"       "study"         "support"
##  [41] "system"        "systems"       "techniques"    "tools"
##  [45] "treatment"     "university"    "various"       "analysis"
##  [49] "applied"       "business"      "can"           "focus"
##  [53] "future"        "help"          "hospital"      "important"
##  [57] "insights"      "intelligence"  "learning"      "management"
##  [61] "may"           "medical"       "methods"       "one"
##  [65] "predictive"    "problem"       "process"       "processing"
##  [69] "related"       "software"      "specific"      "technology"
##  [73] "traditional"   "use"           "used"          "using"
##  [77] "application"   "approach"      "decision"      "patterns"
##  [81] "processes"     "services"      "framework"     "making"
##  [85] "pathway"       "pathways"      "patients"      "activity"
##  [89] "cancer"        "development"   "large"         "model"
##  [93] "stage"         "apache"        "figure"        "methodologies"
##  [97] "source"        "types"         "design"        "challenges"
## [101] "conference"    "engineering"   "international" "journal"
## [105] "proposed"      "review"        "vol"           "discovery"
## [109] "descriptive"   "diagnostic"    "tier"          "solution"
## [113] "solutions"     "studies"       "relevant"      "visualization"
## [117] "jan"           "articles"      "literature"    "dsr"
## [121] "sample"        "qualitative"   "ovarian"
```

```r
#Converting the TDM to an matrix form
m <- as.matrix(tdm)
freq_Count <- sort(rowSums(m),decreasing = T)  #counting the term frequency
head(freq_Count,20)   #first 20  frequency counts
```

```
##        data  healthcare   analytics      health         big information
##         624         339         313         173         163         155
##    research    analysis         can        care     patient      design
##         130         120         100          95          91          91
##        used     systems     process     medical    clinical      system
##          87          84          83          79          76          69
##    patients         vol
##          58          57
```

5. Word Cloud

```r
#5. Create a word cloud with and without color:
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```r
par(mar = c(2, 3, 2, 3)) #(bottom, left, top, right margins)

#Without color
wordcloud(words = names(freq_Count), freq = freq_Count, min.freq = 5, random.order = FALSE)
```


```
#With color
wordcloud(words = names(freq_Count), freq = freq_Count, min.freq = 5,
          random.order = FALSE, colors = rainbow(length(freq_Count)))
```

6. Network Graph

```
#6. Create a network graph:
library(graph)
```

```
## Loading required package: BiocGenerics
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following object is masked from 'package:NLP':
##
##     annotation
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

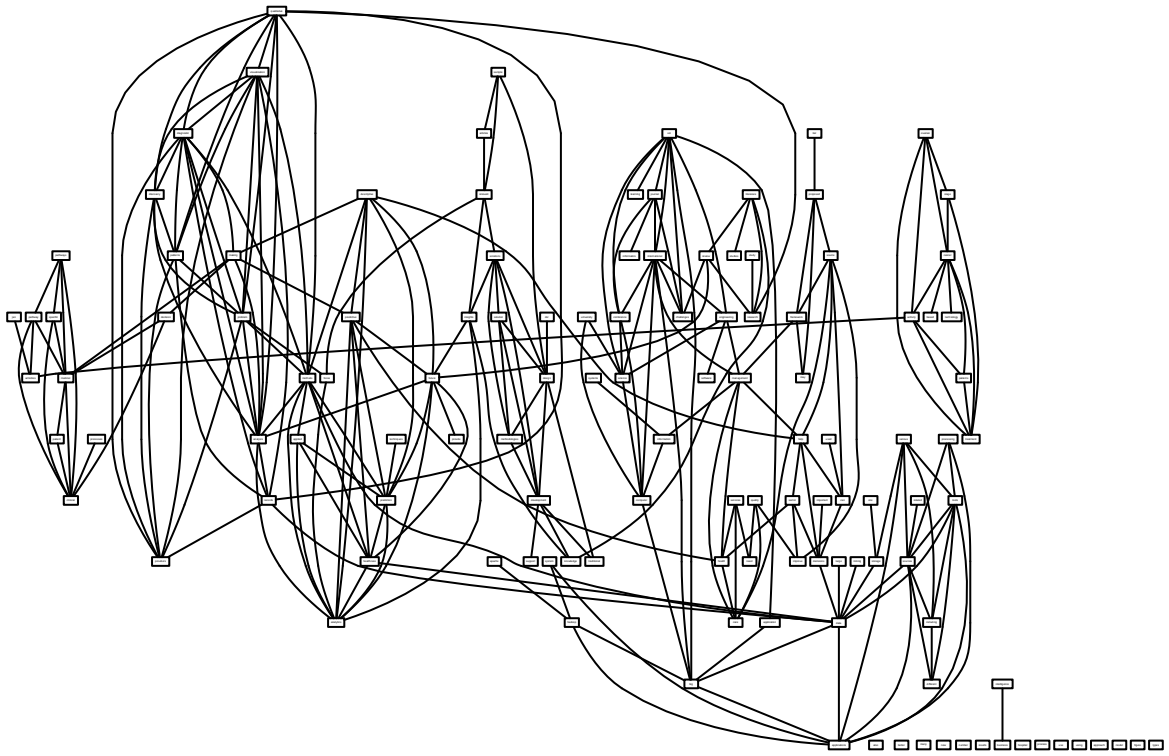
```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
```

```
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min
```

```
library(Rgraphviz)
```

```
## Loading required package: grid
```

```
plot(tdm, term = freq_terms, corThreshold = 0.5)
```



7. Topic modeling

```
#7. Topic modeling
library(tm)
library(topicmodels)
set.seed(07)

dtm <- as.DocumentTermMatrix(t(tdm), weighting = weightTf)
lda_Model <- LDA(dtm, k=5)


#getting the terms in the topic model
terms(lda_Model, 5)
```

```
##      Topic 1      Topic 2       Topic 3      Topic 4     Topic 5
## [1,] "data"       "data"        "data"       "cancer"    "data"
## [2,] "healthcare" "analytics"   "analytics"  "treatment" "health"
## [3,] "analytics"  "healthcare"  "healthcare" "process"   "information"
## [4,] "analysis"   "information" "design"     "medical"   "healthcare"
## [5,] "health"     "research"    "big"        "patients"  "big"
```