Data Visualisation Final Assignment

Riti Chakraborty - 17231417

18 April 2018

Data Visualization Assignment

The main objective of the report is to inform the reader about the topic composition in the corpus provided. It is said that the topics may have a hierarchical structure. Some visualization techniques after fitting clustering algorithms have been used below to gain insights about the data provided and analyse the structure of the topics.

About the Data

There are 7142 text documents in the corpus. These are stored in 19 different folders. However, the folders do not represent the class to which each document belong to. In order to cluster the documents into their relevant groups, I have used two algorithms below to show if the documents can be clustered correctly. Before we start with execution of the algorithm, we need to preprocess the data. The data preprocessing steps are as mentioned below.

```
# Including the required libraries here.
library(tm)
```

```
## Warning: package 'tm' was built under R version 3.4.4
```

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

```
library(SnowballC)
library(wordcloud)
```

```
## Warning: package 'wordcloud' was built under R version 3.4.4
```

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

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.3
```

```
##
## Attaching package: 'ggplot2'
```

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

```
library(ggdendro)
```

```
## Warning: package 'ggdendro' was built under R version 3.4.4
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(cluster)
```

```
## Warning: package 'cluster' was built under R version 3.4.4
```

```
library(HSAUR)
```

```
## Warning: package 'HSAUR' was built under R version 3.4.4
```

```
## Loading required package: tools
```

```
library(fpc)
```

```
## Warning: package 'fpc' was built under R version 3.4.4
```

```
library(skmeans)
```

```
## Warning: package 'skmeans' was built under R version 3.4.4
```

```
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 3.4.4
```

```
## ------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
library(RColorBrewer)
library(gplots)
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:wordcloud':
##
##     textplot
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
library(stats)

options(warn=-1)
```

Reading the corpus

The snippets of code mentioned below deals with reading in the corpus in R and Preprocessing it in order to clean the data and make it appropriate for fitting in clustering algorithms. Vcorpus creates a volatile copy of the corpus read from the dictionary. This volatile copy can be accessed for further processing.

```
#Reading in the corpus from the directory.VCorpus() creates a volatile corpors of the corpus
 in the directory and store it in the environment.
corpus <- VCorpus(DirSource("C:\\Users\\Riti Chakraborty\\Desktop\\corpus_n_topics3", recursi
ve = TRUE, encoding = "UTF-8"), readerControl = list(language = "eng"))
```

Now, we try to look into the contents of the corpus that is read from above. The summary() function displays the name of the document, the type of the document the the mode in which the data is stored in it. The data stored here is in the form of list. The inspect() displays the total number of characters in the first document of the corpus. writeLines() is used to display the contents of the document. It can be observed that, the data stored in here is in the form of list.

```
#creating a backup of the corpus
corpus_backup<-corpus
corpus<-corpus_backup
#displaying the summary of the first five documents in the corpus
summary(head(corpus,5))
```

```
##        Length Class             Mode
## doc1   2      PlainTextDocument list
## doc10  2      PlainTextDocument list
## doc100 2      PlainTextDocument list
## doc101 2      PlainTextDocument list
## doc102 2      PlainTextDocument list
```

```
#inspecting first doc of the corpus
inspect(corpus[1])
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 1
##
## [[1]]
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 926
```

```
#displaying the content of the first doc in the corpus
writeLines(as.character(corpus[1]))
```

```
## list(list(content = c("From: decay@cbnewsj.cb.att.com (dean.kaflowitz)", "Subject: Re: abo
ut the bible quiz answers", "Organization: AT&T", "Distribution: na", "Lines: 18", "", "In ar
ticle <healta.153.735242337@saturn.wwc.edu>, healta@saturn.wwc.edu (Tammy R Healy) writes:",
"> ", "> ", "> #12) The 2 cheribums are on the Ark of the Covenant.  When God said make no ",
"> graven image, he was refering to idols, which were created to be worshipped. ", "> The Ark
of the Covenant wasn't wrodhipped and only the high priest could ",
## "> enter the Holy of Holies where it was kept once a year, on the Day of ", "> Atonemen
t.", "", "I am not familiar with, or knowledgeable about the original language,", "but I beli
eve there is a word for \"idol\" and that the translator", "would have used the word \"idol\"
instead of \"graven image\" had", "the original said \"idol.\"  So I think you're wrong here,
but", "then again I could be too.  I just suggesting a way to determine", "whether the interp
retation you offer is correct.", "", "",
## "Dean Kaflowitz"), meta = list(author = character(0), datetimestamp = list(sec = 5.4517250
0610352, min = 21, hour = 22, mday = 19, mon = 3, year = 118, wday = 4, yday = 108, isdst =
0), description = character(0), heading = character(0), id = "doc1", language = "eng", origin
= character(0))))
## list()
## list()
```

Data Preprocessing

In this step I am removing all the numbers and special characters. Also, the document has been stemmed (words are converted into its root form), it has been converted into lower case, digits and single letters have been removed and finally even the stop words have been removed.

Importance of this step: Computer cannot always read punctutation and other special characters as it is and treat these special characters more as a word. This might lead to ambiguous results. Therefore, it is better to remove the punctuations or special characters from the text before analysis.

Creation of Document Term Matrix

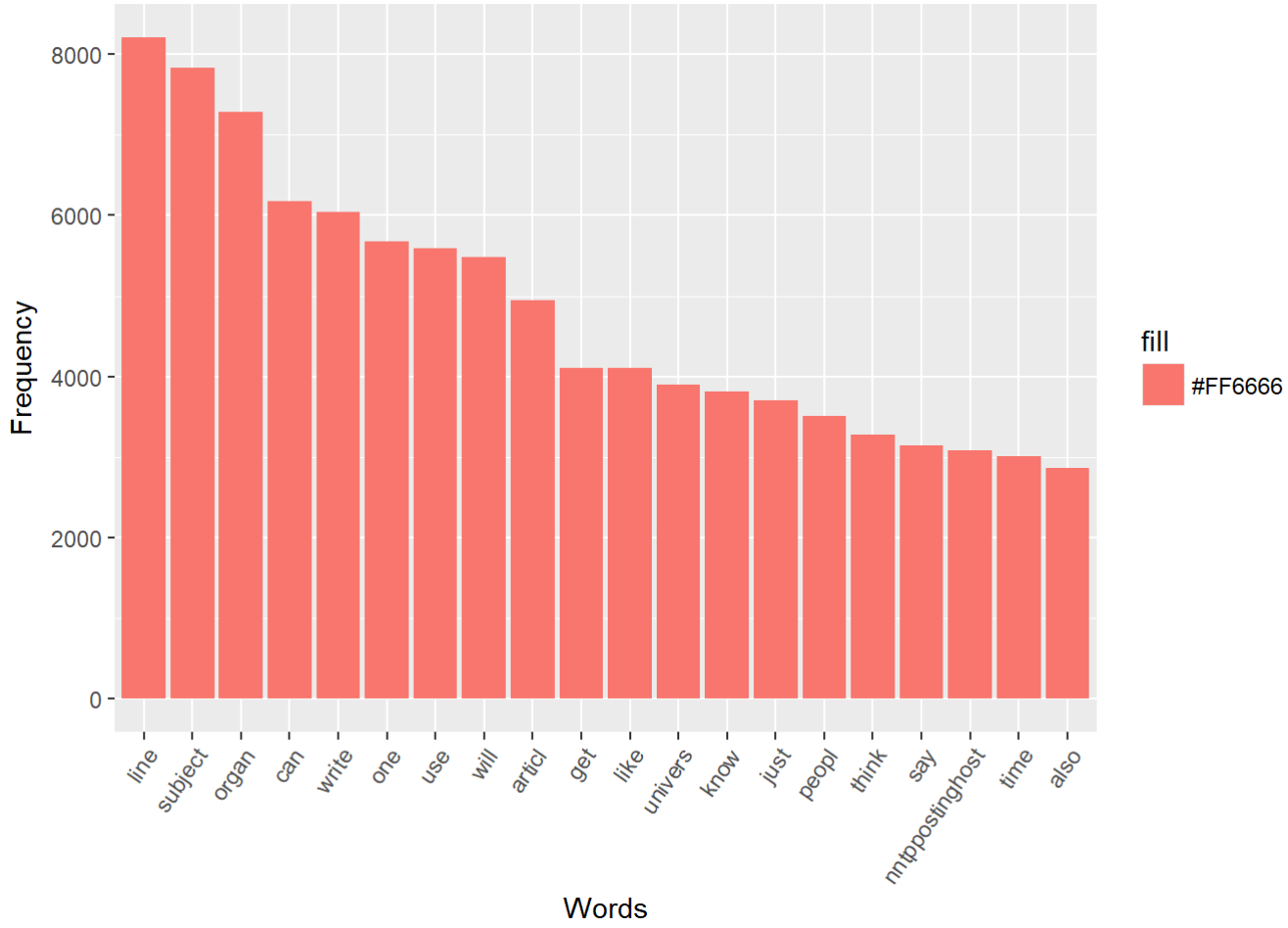
Creating a document term matrix. The matrix contains the term frequency. The function removeSparseTerms() is used to removing the sparse rows.

```
corpus.dtm <- DocumentTermMatrix(corpus)
#corpus.dtm <- DocumentTermMatrix(corpus, control = list(weighting = function(x) weightTfIdf
(x, normalize = TRUE)))
corpus.dtm<-removeSparseTerms(corpus.dtm, 0.999)
```

Displaying Word Frequency

The bar chart below shows the most frequently used words in the sample data. It can be observed that line, subject and organ are the three most frequently used words.

```
#Converting the DTM to a matrix form
corpus.dtm.mat <- corpus.dtm %>% as.matrix()

#Extracting the frequncy column
freq <- colSums(as.matrix(corpus.dtm))

#Creating a dataframe with word and its frequency
wf <- data.frame(word=names(freq), freq=freq)

#Creating a new dataframe with word and frequency stored in descending order.
new_wf <- wf[order(-freq),]

#creating backup
wf1_1<-new_wf

#Plotting word frequency barchart
p <- ggplot(head(wf1_1,20), aes(x = reorder(word, -freq), y = freq, fill = "#FF6666")) +
  geom_bar(stat = "identity")+ xlab("Words")+ ylab("Frequency")+
  theme(axis.text.x=element_text(angle=55, hjust=1))
p
```



Fitting the skmeans algorithm

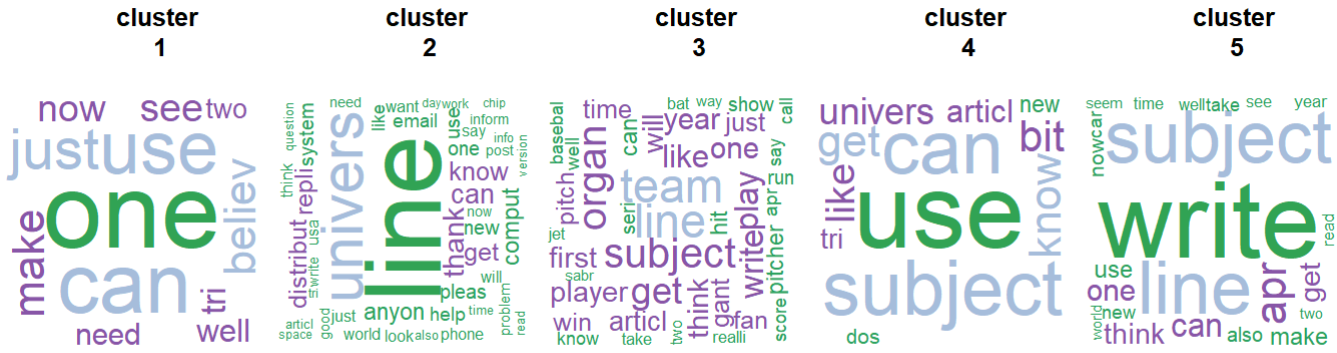
skmeans() is a function already defined in the library which is used to fit a statistical Kmeans algorithm. skmeans() uses cosine similarity to find the distances between vectors and the centroid. The clusters formed are stored in the form of vector and converted into dataframe for visualisation purpose.

```
#This part of the code is for sampling the data before fitting the model. But, I have already
 taken a sample of the main corpus at the start there here i am using 100% of the sample dat
a.

percent = 20
sample_size = nrow(corpus.dtm.mat) * percent/100

#extract rows from the DTM after sampling
corpus.dtm.mat.sample <- corpus.dtm.mat[sample(1:nrow(corpus.dtm.mat), sample_size, replace=F
ALSE),]

#declaring number of clusters
k=5

#call the skmeans function it returns a vector of cluster assignments
corpus.dtm.mat.sample.skm  <- skmeans(corpus.dtm.mat.sample,k, method='genetic')

# Converting the vector to a data frame and renaming the columns
corpus.dtm.mat.sample.skm <- as.data.frame(corpus.dtm.mat.sample.skm$cluster)
colnames(corpus.dtm.mat.sample.skm) = c("cluster")

#Storing Rownames as a column
corpus.dtm.mat.sample.skm$docs <- rownames(corpus.dtm.mat.sample.skm)

# I unlist the list assigned by rownames to $docs
corpus.dtm.mat.sample.skm$docs <- unlist(corpus.dtm.mat.sample.skm$docs)
corpus.dtm.mat.sample.skm.table <-table(corpus.dtm.mat.sample.skm$cluster, corpus.dtm.mat.sam
ple.skm$docs)
```

Visualising skmeans with the help of word clouds

The clusters formed using the skmeans algorithm are plotted as word Cloud using the code mentioned below.

```
#Visualising skmeans output with the help of word cloud.

#converting table into a dataframe
corpus.dtm.mat.sample.skm.table <-as.data.frame.table(corpus.dtm.mat.sample.skm.table)

#creating term document matrix
corpus.tdm <- TermDocumentMatrix(corpus, control = list(weighting = function(x) weightTf(x)))

#Removing Sparse terms
corpus.tdm<-removeSparseTerms(corpus.tdm, 0.999)

# select only the documents from the  random sample taken earlier
corpus.tdm.sample <- corpus.tdm[, rownames(corpus.dtm.mat.sample)]

# convert to r matrix
corpus.tdm.sample.mat <- corpus.tdm.sample %>% as.matrix()

# number of clusters
m<- length(unique(corpus.dtm.mat.sample.skm$cluster))

#For layout of the matrix
par(mfrow=c(2,m))

# for each cluster plot an explanatory word cloud
for (i in 1:m) {
  #the documents in  cluster i
  cluster_doc_ids <-which(corpus.dtm.mat.sample.skm$cluster==i)

  #the subset of the matrix with these documents
  corpus.tdm.sample.mat.cluster<- corpus.tdm.sample.mat[, cluster_doc_ids]

  # sort the terms by frequency for the documents in this cluster
  v <- sort(rowSums(corpus.tdm.sample.mat.cluster),decreasing=TRUE)
  d <- data.frame(word = names(v),freq=v)
  # call word cloud function
  wordcloud(words = d$word, freq = d$freq, scale=c(5,.2), min.freq = 3,
            max.words=60, random.order=FALSE, rot.per=0.35,
            colors=c('#2ca25f','#8856a7','#a6bddb','#a6bddb','#31a354'))
  title(paste("cluster\n", i))
}
```



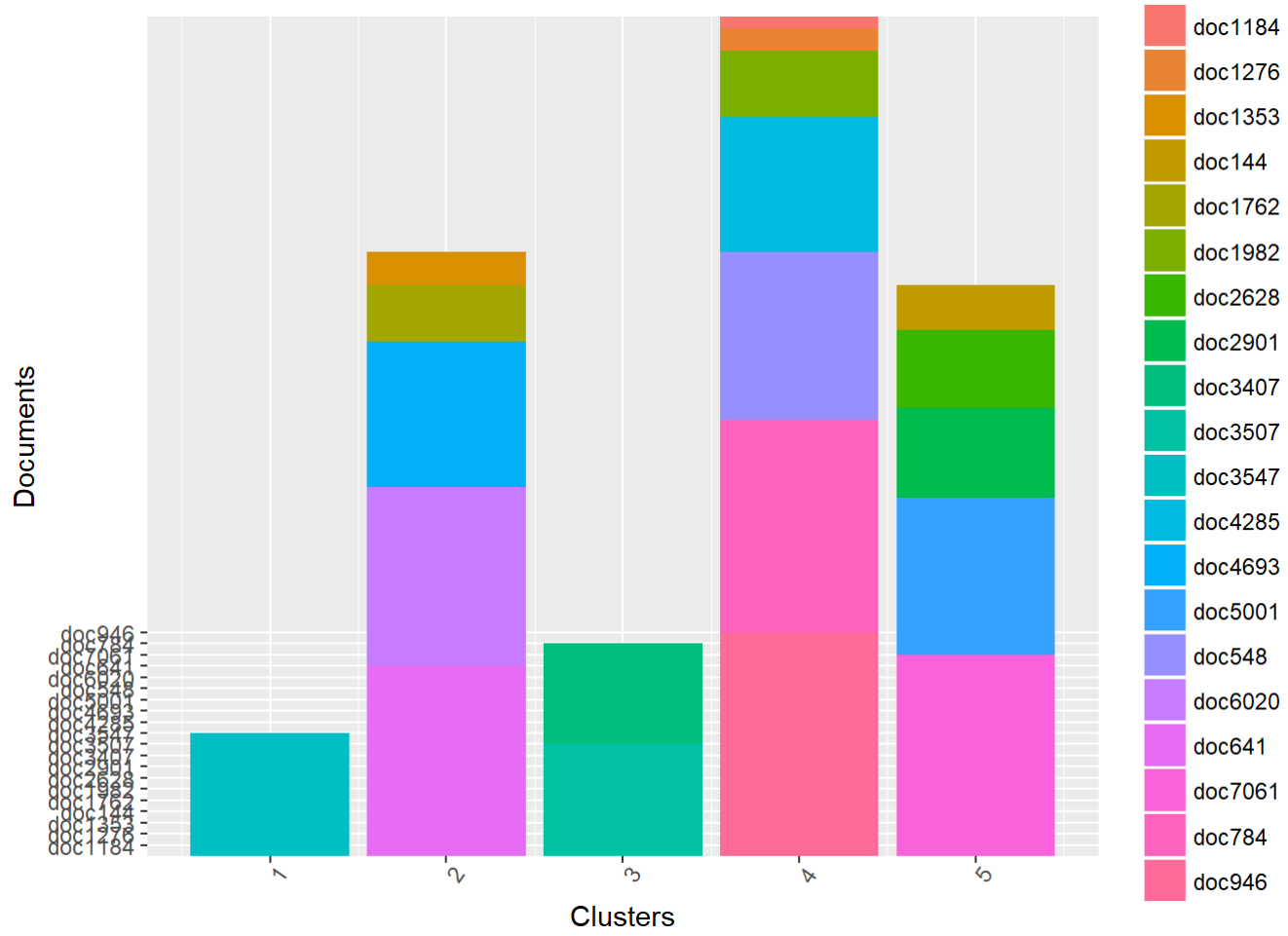
Observation made from the word cloud above:

1. The clusters are not clearly distinct. The reason behind this could be that the documents indeed have a hierarchical structure.
2. There are few words which appear in each of the clusters. Some of the words which might be relevant to each other with respect to meaning or general usage are visible in different clusters.
3. This shows that the cluster boundaries are not clean and there is overlap between clusters.

Plotting the cluster composition

The bar graph below can be used to see the composition of each cluster generated from the model above. The bar chart is plotted with top 20 records of the data stored in 'corpus.dtm.mat.sample.skm'. This table contains the assigned cluster to each document that is being trained.

```
library(scales)
#corpus.dtm.mat.sample.skm
#plot(corpus.dtm.mat.sample.skm)
p_1<- ggplot(head(corpus.dtm.mat.sample.skm,20), aes(x = cluster, y = docs, fill = docs)) +
  geom_bar(stat = "identity")+ xlab("Clusters")+ ylab("Documents")+
  theme(axis.text.x=element_text(angle=55))

p_1
```



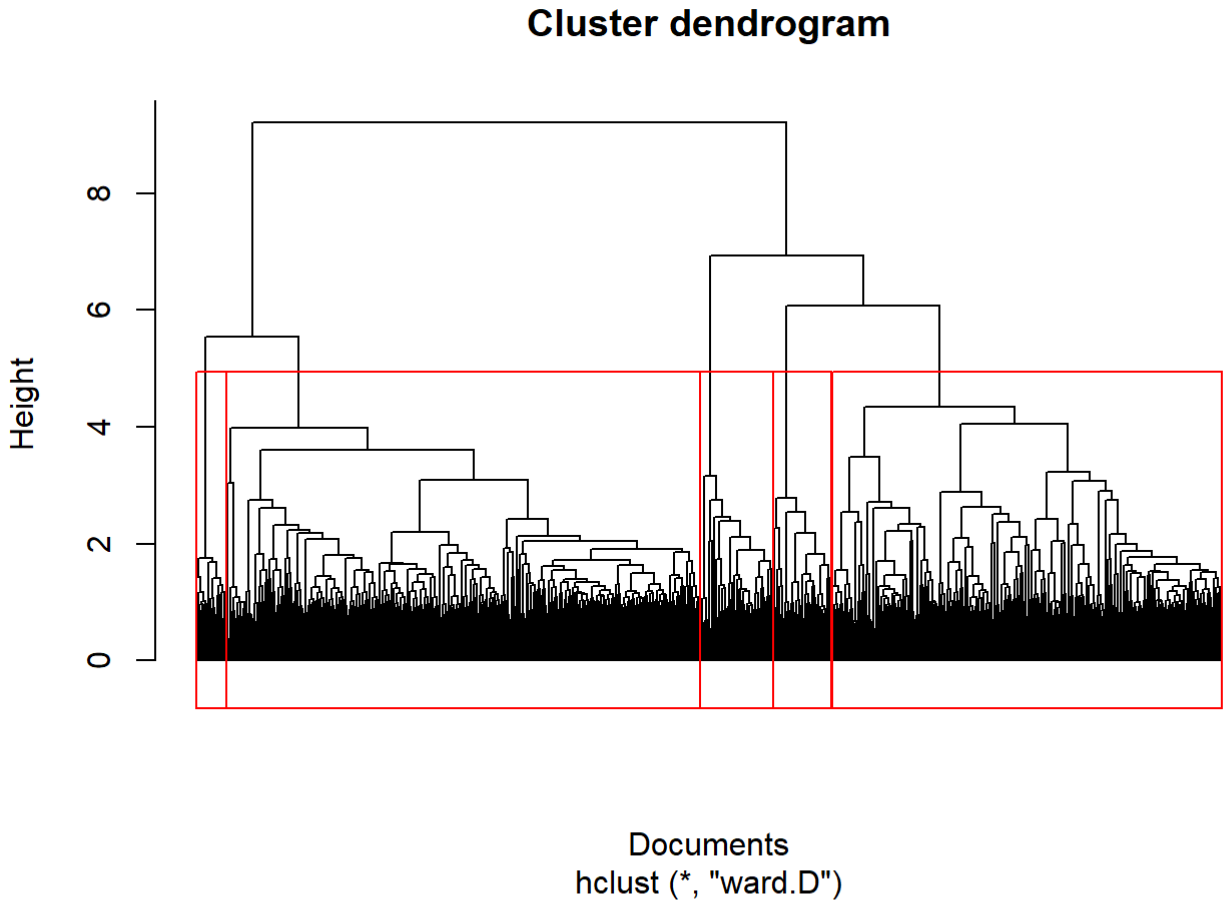
Observation made from the Bar Chart :

1. Cluster 2: seems to be more pure that the other four clusters. This is because it 4 different type of documents in it (visible) from the four different types of colors present.
2. The clusters are mostly heterogenous. Thus, we move on to hierarchical clustering to check if the results are better.

Performing Hierarchical Clustering.

I am using wards method for hierarchocal cluatering here. Ward's method minimizes the total cluster variance within a cluster. At each step the pair of clusters with minimum between-cluster distance are merged together. Clusters are formed in a manner that minimizes the loss of data associated with each clustering.

```
# philentropy library provides a number of distance/similarity measures, including cosine whi
ch we use for group documents
library(philentropy)

#From philentropy library. Slower than dist function, but handles cosine similarity
sim_matrix<-distance(corpus.dtm.mat.sample, method = "cosine")

# for readiblity (and debugging) put the doc names on the cols and rows
colnames(sim_matrix) <- rownames(corpus.dtm.mat.sample)
rownames(sim_matrix) <- rownames(corpus.dtm.mat.sample)

# cosine is really a similarity measure (inverse of distance measure)
# we need to create a distance measure for hierarchical clustering
dist_matrix <- as.dist(1-sim_matrix)

# hierarchical clustering
corpus.dtm.sample.dend <- hclust(dist_matrix, method = "ward.D")

# plot the dendogram
par(mfrow=c(2,1))
plot(corpus.dtm.sample.dend, hang= -1, labels = FALSE,  main = "Cluster dendrogram", sub = NU
LL, xlab = "Documents", ylab = "Height")

# here rect.hclust creates  rectangles around the dendogram for k number of clusters
rect.hclust(corpus.dtm.sample.dend, k = 5, border = "red")

#Cutting the tree

# number of clusters we wish to examine
k=5

# call the cutree function, cutree returns a vector of cluster membership in the order of the
 original data rows
corpus.dtm.sample.dend.cut <- cutree(corpus.dtm.sample.dend, k=5)

#number of clusters at the cut
m <- length(unique(corpus.dtm.sample.dend.cut))

# create a data frame from the cut
corpus.dtm.sample.dend.cut <- as.data.frame(corpus.dtm.sample.dend.cut)

#add a meaningful column namane
colnames(corpus.dtm.sample.dend.cut) = c("cluster")

# add the doc names as an explicit column
corpus.dtm.sample.dend.cut$docs <- rownames(corpus.dtm.sample.dend.cut)

#Reformatting the dataframe
corpus.dtm.sample.dend.cut$docs <- unlist(corpus.dtm.sample.dend.cut$docs)

# create a frequency table
corpus.dtm.sample.dend.cut.table <-table(corpus.dtm.sample.dend.cut$cluster, corpus.dtm.sampl
e.dend.cut$docs)

#displays the confusion matrix
#corpus.dtm.sample.dend.cut.table
```

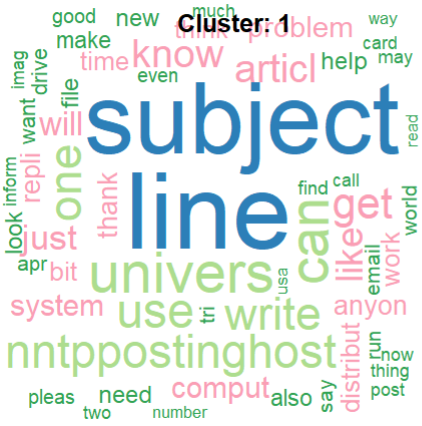


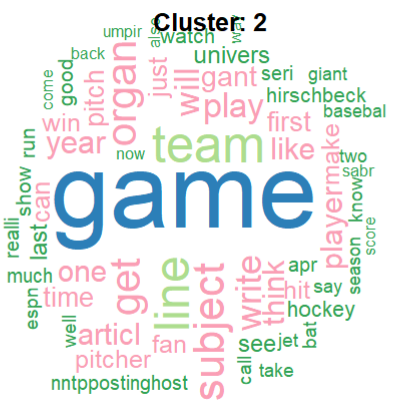
Observations (Dendogram)

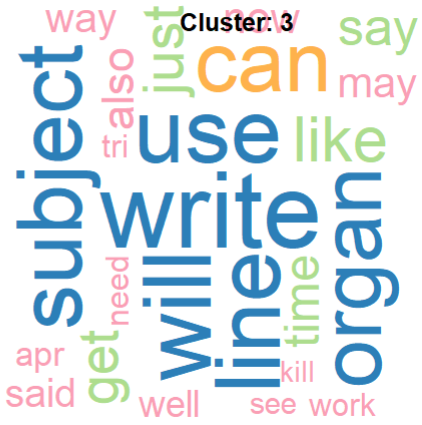
1. We can that the corpus indeed has a hierarchical structure.

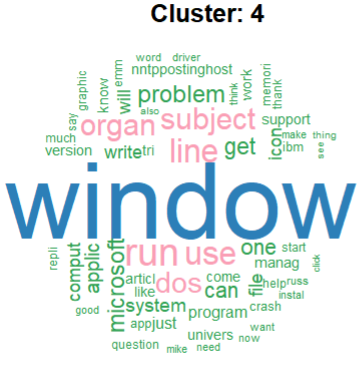
2. Their are approximately 5 subtrees visible, connected to a root.
3. For inspecting the composition of each cluster a.k.a subtree, i am using word cloud below.
4. For every cluster a word cloud is created. This helps in looking into the composition of the cluster i.e. words contained in each.

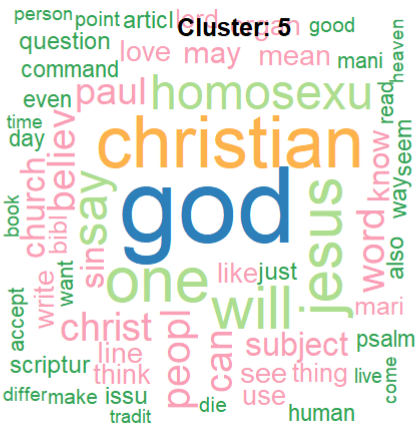
```
#Visualising Hclust() output with the help of word cloud
#number of clusters at the cut
m <- length(unique(corpus.dtm.sample.dend.cut$cluster))

#Layout of matrix
par(mfrow=c(2,3))

# for each cluster plot an explanatory word cloud
for (i in 1:m) {
  #the documents in  cluster i
  cut_doc_ids <-which(corpus.dtm.sample.dend.cut$cluster==i)

  #the subset of the matrix with these documents
  corpus.tdm.sample.mat.cluster<- corpus.tdm.sample.mat[, cut_doc_ids]

  # sort the terms by frequency for the documents in this cluster
  v <- sort(rowSums(corpus.tdm.sample.mat.cluster),decreasing=TRUE)
  d <- data.frame(word = names(v),freq=v)
  # call word cloud function
  wordcloud(words = d$word, freq = d$freq, scale=c(5,.2), min.freq = 3, max.words=60, random.
order=FALSE, rot.per=0.35,
            colors=c('#31a354','#fa9fb5','#addd8e','#feb24c','#2c7fb8'))
  title(paste("Cluster:", i))
}
```











Observation:

(the cluster position may vary therefore word cloud 1 might not always address to cluster 1 but it will sddress to one of the clusters)

From the above picture, the topics contained in this corpus is somewhat clear. Wordclouds displays the frequently used terms in a cluster(here!) to explain its composition. The size represents the most frequently used words.

Topic composition of the corpus ~ Topic composition of the word clouds.

There are five clusters created therefore there are five topics which are part of the topics in the corpus. Topic 1: One of the clusters contains words like turkish people, armenia etc which hints to the fact that the corpus contains certain topics related to a specific geograophic region or people belonging to that region.

Topic 2: One of the clusters conatins words like subject, line, write, compute, jpeg, people, work, article etc. which deals with ‘computer related work or media’.

Topic 3: One of the clusters has words like drive, subject, disk, dos, read get, icon, control etc. deals with certain kinds of association(group of people working for a cause). One of the words in this wordcloud is SCSI which is a professional body for construction, land and property in Ireland. Thus, I am assuming that this cluster refers to similar topics.

Topic 4: This has words related to beliefs or religion in general. Presence of words like God which is the frequent word in the cluster shows the relevance more. Also words like, jesus, sin, christian, christ, psalm, church, faith etc. besides words like homosexual, love etc. reflect the genre of this cluster. Therefore we can say that the corpus contains related topics as well.

Topic 5:The last cluster seems pretty straight forward. It has words like game, pitcher,umpire,guy, playoff, goal etc. This reflect that the composition of the cluster may be topics related to Sports or games or outdoor activities.

(note: some of the word appeared in the word cloud but are not visible on the rmd version)

From the above explanation about topic composition of each cluster, it can be concluded that the corpus might deal with atleast 5 topics and they are ‘About a country’, ‘About media or study’, ‘About professional/government body working for some cause’, ‘About Religion and Beliefs’ and Finally about ‘Sports or Games’