Health News in Twitter Data Set

1 Exercise

Each file is related to one Twitter account of a news agency. For example, “bbchealth.txt” is related to BBC health news. Each line contains “tweet id”, “date and time”, “tweet”. The separator is '|'.

This text data has been used to evaluate the performance of topic models on short text data. However, it can be used for other tasks such as clustering.

1.1 Overview

The target is to investigate the provided data set by an unsupervised learning approach for exploring the most frequent terms in every data frame.

In general, the whole procedure of the created R-script concludes the following steps:

- Load the needed text file into a data frame
- Clean the data (Preprocessing)
- Build corpus
- Create a term matrix
- Apply hierarchical word clustering by a dendrogram
- Apply nonhierarchical k-means clustering

2 Procedure

2.1 Cleaning, corpus, term matrix

With the code from the script-part {r setup} the following steps are being performed:

- Needed working library is loaded
- The current output of the dataset can be checked with the help of the Console. Type in “tweets” to print its content.
- There’s already an output visible but we’d like to focus on the content of the tweet and not its “ID” or “timestamp”. Therefore, we perform some cleaning procedures within this part:
- Being able to create a matrix of the most frequent terms within our next step, we first need to create a so-called “corpus” to store all the cleaned tweets:
- After creating the corpus, the “term matrix” is being created. Also, some restrictions can be set. For example, that the defined word length of any term that should be counted has to be between “1” and “*N*”.

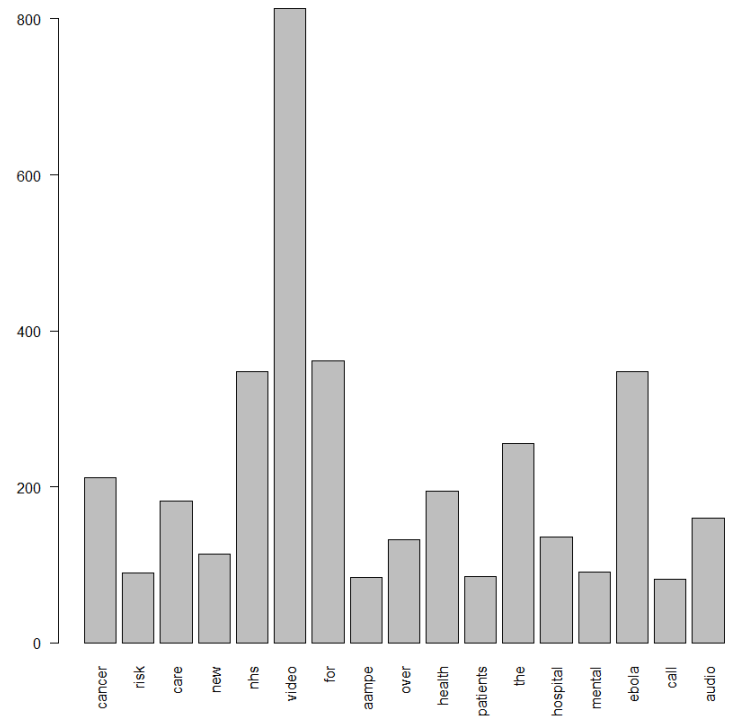
```r
```{r read}
library(tm)
library(cluster)
library(readr)

# Read the file you want to analyze, make sure the Text Mining library is installed
# !! Make sure to set your working directory first! --> setwd("C:/Users/xx/your_folderpath")

tweets <- readLines("Health-News-Tweets/bbchealth.txt")
```

```{r setup}

#Build the corpus
corpus <- Corpus(VectorSource(tweets))
removeURL <- function(x) gsub("http://([[:alnum:]|[:punct:]])+", "", x)
corpus <- tm_map(corpus, content_transformer(removeURL))
#corpus <- tm_map(corpus, content_transformer(tolower))
# remove punctuation
corpus <- tm_map(corpus, removePunctuation)
# remove numbers
corpus <- tm_map(corpus, removeNumbers)
# add extra stop words for example 'available' or 'via'
myStopwords <- c("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun", "Jan", "Feb", "Mar", "Apr", "May",
#myStopwords <- c("mon", "tue", "wed", "thu", "fri", "sat", "sun", "jan", "feb", "mar", "apr", "may",
# remove stopwords from corpus
corpus <- tm_map(corpus, removeWords, myStopwords)

#Create a matrix related to the terms. Set the minimum wordlength to 1 until Infinity
TermMatrix <- TermDocumentMatrix(corpus, control = list(minWordLenght=c(1, Inf)))
t <- removeSparseTerms(TermMatrix, sparse = 0.98)
m <- as.matrix(t)
```
```

Within our next step, we'll finally create a plot by using the created matrix from before. The following code creates a barplot from all the terms which are at least 30 times mentioned within our matrix.

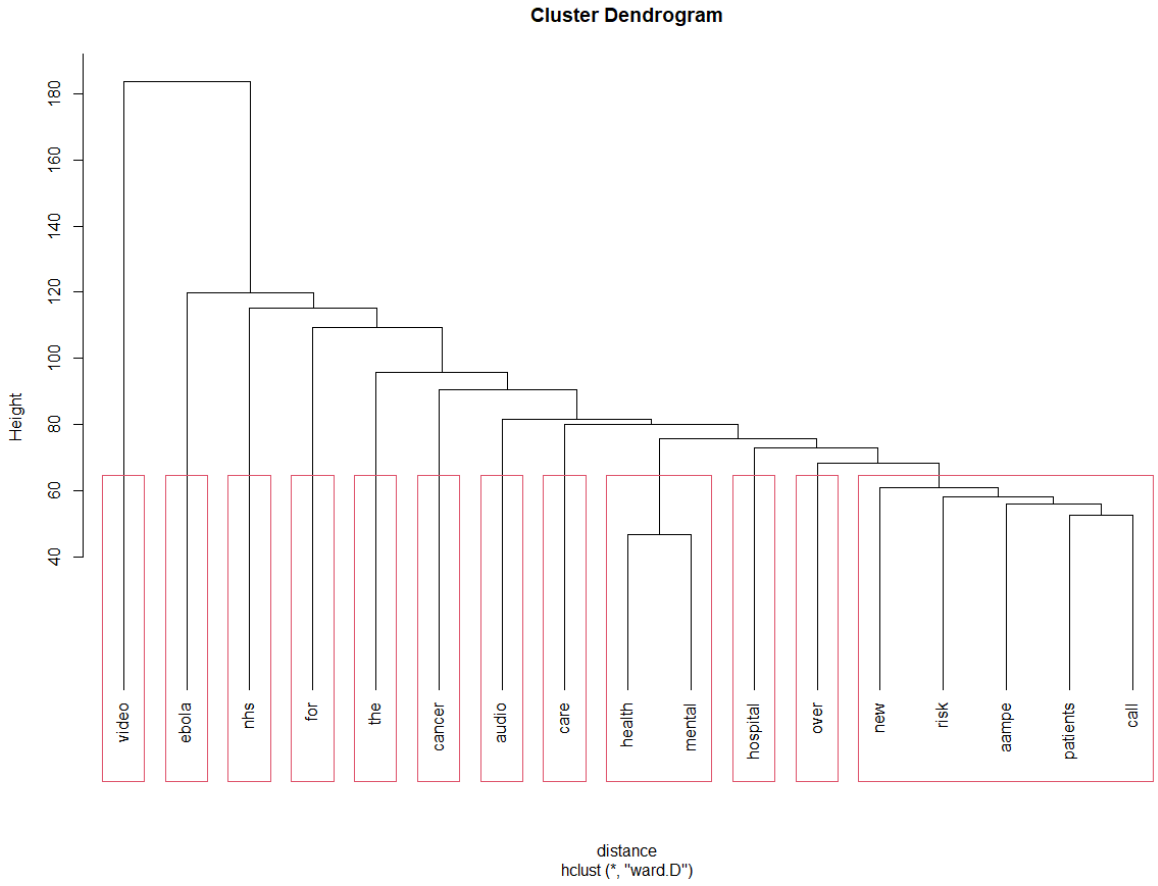


We can see, that within the file "bbchealth.txt" besides often used articles and pronouns also the terms "cancer", "care" or "ebola" are being used.

2.2 Creating a dendrogram

Now it helps us to create a dendrogram to recognize which terms might show up in the same cluster. See for that the yellow marked values as examples:

- The lower the value between two terms is → the higher the possibility these two terms might show up in the same cluster. There might be some relationship.
- The higher the value → the lower the possibility that these terms are assigned in the same cluster.



Within this plot, you can see that further preprocessing would be needed for getting a better result as there are still some words that are "hanging around". Nevertheless, the setting of 12 clusters "*k*" seems to be an acceptable result for this task.

2.3 Create nonhierarchical

In our last step, a nonhierarchical cluster by using k-means is being performed.
Within the output, you can see all the 12 listed clusters and their sizes. Furthermore, you can see the averages of every investigated term per cluster.

For example:
The term “cancer” has an average of “0.072” that it appeared within “cluster 3”, an average of “0.064” that it appeared in “cluster 4”, etc.

High average → high frequency of appearance within the cluster.

```
> print(kc)
K-means clustering with 12 clusters of sizes 200, 301, 250, 157, 2179, 104, 162, 19, 53, 285, 118, 101

Cluster means:
       cancer        risk        care         new         nhs       video         for       aampe        over      health    patients         the    hospital
1  0.00000000 0.005000000 0.025000000 0.00500000 0.135000000 0.2650000 0.07000000 0.00500000 0.005000000 0.01000000 0.000000000 1.100000000 0.005000000
2  0.00000000 0.016611296 0.003322259 0.00000000 0.006644518 0.2790698 0.09966777 0.00000000 0.003322259 0.01661130 0.009966777 0.000000000 0.009966777
3  0.07200000 0.028000000 0.104000000 0.00000000 0.008000000 0.2080000 1.01600000 0.01600000 0.000000000 0.02000000 0.028000000 0.004000000 0.016000000
4  0.06369427 0.019108280 0.031847134 0.02547771 0.044585987 0.0000000 0.08280255 0.03821656 0.044585987 0.03184713 0.006369427 0.012738854 0.025477707
5  0.00000000 0.025240936 0.053235429 0.00000000 0.000000000 0.2005507 0.00000000 0.02937127 0.000000000 0.00000000 0.025699862 0.000000000 0.048187242
6  0.05769231 0.000000000 0.009615385 1.00000000 0.182692308 0.2019231 0.15384615 0.01923077 0.009615385 0.01923077 0.019230769 0.048076923 0.019230769
7  1.00000000 0.043209877 0.018518519 0.00000000 0.000000000 0.2160494 0.00000000 0.00000000 0.000000000 0.00000000 0.043209877 0.018518519 0.037037037
8  0.00000000 0.000000000 0.000000000 0.00000000 0.000000000 0.3684211 0.21052632 0.00000000 0.000000000 0.00000000 0.000000000 1.052631579 0.000000000
9  0.01886792 0.018867925 0.056603774 0.03773585 0.037735849 1.0377358 0.05660377 0.00000000 0.056603774 0.94339623 0.056603774 0.056603774 0.018867925
10 0.04210526 0.007017544 0.038596491 0.00000000 1.003508772 0.2000000 0.08070175 0.02105263 0.059649123 0.01052632 0.014035088 0.000000000 0.010526316
11 0.00000000 0.042372881 0.059322034 0.01694915 0.025423729 0.0000000 0.01694915 0.00000000 0.008474576 1.00000000 0.008474576 0.008474576 0.008474576
12 0.02970297 0.029702970 0.039603960 0.00990099 0.000000000 0.1188119 0.01980198 0.00990099 1.000000000 0.03960396 0.009900990 0.000000000 0.049504950
       mental       ebola        call       audio
1  0.010000000 0.00000000 0.010000000 0.005000000
2  0.000000000 1.00000000 0.003322259 0.000000000
3  0.004000000 0.01200000 0.144000000 0.004000000
4  0.006369427 0.01273885 0.000000000 1.000000000
5  0.000458926 0.00000000 0.014685636 0.000000000
6  0.000000000 0.06730769 0.019230769 0.000000000
7  0.000000000 0.00000000 0.018518519 0.000000000
8  0.000000000 1.00000000 0.000000000 0.000000000
9  0.509433962 0.00000000 0.000000000 0.000000000
10 0.000000000 0.01754386 0.003508772 0.000000000
11 0.491525424 0.00000000 0.016949153 0.008474576
12 0.000000000 0.10891089 0.019801980 0.000000000
```