Task0 : Understanding the problem

Jean-Luc BELLIER

23 avril 2017

General purpose

The goal of this task is to load a set of files and getting familiar with some general features of these files : general properties, and extraction of some words from these files using regular expressions.

Preliminaries : load libraries

```
library("clue")
```

```
## Warning: package 'clue' was built under R version 3.3.3
```

```
library("tm")
```

```
## Warning: package 'tm' was built under R version 3.3.3
```

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

```
## Warning: package 'NLP' was built under R version 3.3.2
```

```
script.dir <- "C:\\perso\\Culture generale\\Coursera\\Data_Science\\Cours_10_Data_Science_Capstone\\fina
```

File load

The files come from a dataset of files provided by the Capstone Dataset. This dataset gathers 3 types of data : blogs, tweets and news, for 4 different languages : German, Finnish, English and Russian.

For this file, we only focus on the english dataset.

```
# First we get the folder of the current script
#script.dir <- dirname(sys.frame(1)$ofile)
#print(script.dir)
inputfiles.dir <- file.path(script.dir,"en_US")
print(inputfiles.dir)
```

```
## [1] "C:\\perso\\Culture generale\\Coursera\\Data_Science\\Cours_10_Data_Science_Capstone\\final/en_US
```

```
# Here we define a function to get the general properties of the file. The full path to the file is an

fileInformation <- function(filepath)
{
    # First we build a connection to th input file
    con = file(filepath,open="rb")

    fileRead <- readLines(filepath,skipNul = TRUE,encoding="UTF-8-BOM")

    # compute general parameters : file size, number of lines and length of longest line
    size_MB <- file.info(filepath)$size/1048576
    nbLines <- length(fileRead)
```

```
    maxLineLength <- max(sapply(fileRead,nchar))

    close(con)

    # We build the output dataframe, which gather the main file properties for the input file
   df <- data.frame(basename(filepath),size_MB,nbLines,maxLineLength)
   names(df) <- c("File","size_MB","NbLines","MaxLineLength")

   df
}
```

We can now use this function to determine the parameters of the set of files : size in MB, number of lines and highest length of line (in number of characters).

```
twitter_info = fileInformation(paste(inputfiles.dir,"en_US.twitter.txt",sep="/"))
blogs_info = fileInformation(paste(inputfiles.dir,"en_US.blogs.txt",sep="/"))
news_info = fileInformation(paste(inputfiles.dir,"en_US.news.txt",sep="/"))
```

```
## Warning in readLines(filepath, skipNul = TRUE, encoding = "UTF-8-BOM"):
## ligne finale incomplète trouvée dans 'C:\perso\Culture generale\Coursera
## \Data_Science\Cours_10_Data_Science_Capstone\final/en_US/en_US.news.txt'
```

```
print("General information on input files")
```

```
## [1] "General information on input files"
```

```
df_files <- rbind(twitter_info,blogs_info,news_info)
df_files
```

```
##                File  size_MB NbLines MaxLineLength
## 1 en_US.twitter.txt 159.3641 2360148           213
## 2   en_US.blogs.txt 200.4242  899288         40835
## 3    en_US.news.txt 196.2775   77259          5760
```

Focus on the tweets

Here we will focus on the tweets, to get the tweets corresponding to input keywords.

```
MatchTweets <- function(filepath)
{

    con = file(filepath,open="rb")
    fileRead <- readLines(filepath,skipNul = TRUE,encoding="UTF-16LE")
    # Checks only done for the tweets
    if (grepl("twitter.txt",filepath)) {
        ratio = 0
        # find word "love" and "hate" in the tweets
        match_love <- grep("love",fileRead,perl=TRUE)
        match_hate <- grep("hate",fileRead,perl=TRUE)
        if (length(match_hate)!=0) {
            ratio = length(match_love) /  length(match_hate)
            print(sprintf("ratio love vs hate = %f",ratio))
        }
        # display the twwet related to biostats
        match_biostats <- grep("biostats",fileRead,perl=TRUE,value=TRUE)
        print(match_biostats)
```

```
        match_tweet2 = grep("A computer once beat me at chess, but it was no match for me at kickboxing
        print(sprintf("number of tweets : %d",length(match_tweet2)))
    }
    close(con)
}

GetTweets <- MatchTweets(paste(inputfiles.dir,"en_US.twitter.txt",sep="/"))
```

```
## [1] "ratio love vs hate = 4.108592"
## [1] "i know how you feel.. i have biostats on tuesday and i have yet to study =/"
## [1] "number of tweets : 3"
```

A few notes concerning the data

Where do the data come from ?

The data are issued from HC Corpora, which contains file sets in several languages (english, german, finnish and russian). For each language, we have 3 files : tweets, blogs and news.

What do the data look like?

1. After reading of the files, we can notice that some characters are not standard ASCII characters, and introduice some noise in the texts.
2. Frequent words do not provide much sense to the texts. These words are called 'stopwords'. They are articles, auxiliaries, pronouns, ...

What are the common steps in natural language processing ?

1. Removal of extra blank spaces
2. removal of numbers
3. Removal of stopwords
4. Removal of punctuation signs

What are some common issues in the analysis of text data?

The common issues are the presence of non-standard ASCII characters. Furthermore, texts can contain information in different languages. We need to know which language is the most probable : this is important to get the most probable language, because it will determine which stopwords to remove.

A raw analysis will consist in splitting the text in single words, and counting the number of occurrences of each word in the text. By sorting the number of occurrences in decreasing order, this will give us the words that are most frequently used.

A further analysis will consist in creating words associations. These associations are called NGrams. These NGrams are sequences of N words, that gives us information about the contexts in which the words appear.