Text Mining Lab 2 – Classifying Text

Use this lab time to gain experience with text mining techniques in R / Python. ULTIMATE GOAL: Use text mining strategies to classify e-mails as good e-mails ('ham') or bad e-mails ('spam')

Dataset

SMSSpamCollection.csv

Problem Description

The SMSSpamCollection dataset contains SMS/e-mails classified as either good e-mails ('ham') or bad e-mails ('spam'). The goal is to use this dataset to enable the classification of new e-mails as either ham or spam.

0. Start by **initializing the environment.** With R, you might find the following packages to be useful:

```
library(tm) # for text mining functions
library(qdap) # for some text processing functions
library(e1071) # for Naive Bayes and Support Vector Machines methods
library(dplyr) # for tidyverse processing
library(tidytext) # for tidyverse analysis
library(ggplot2) # for tidyverse plotting
```

1. Load the data and **explore** its characteristics. How can you ensure that the messages are in 'character' format? How many ham messages are there? Spam? Is the length of the message linked to the email category? What visualizations can you provide at this stage?

2. Prepare and transform the data so that it is suitable for input into a classifying algorithm. You can see an example of how to transform the text with the tm package into a corpus in notebook TMNLP 01 (see supplementary material). How will you tackle text cleaning? Will you replace abbreviations, remove punctuation, remove numbers, stem the terms, de-capitalize terms, remove extraneous white spaces, remove stop words, etc. Your ultimate goal should be to create a **document term matrix** on which to train classifiers.

3. Select an appropriate classifier and train it using the data (this will require you to create a representative training/testing pair). Another issue to consider is the sparsity of the DTM: is it worth keeping every token? Perhaps only those tokens appearing in more than *x* messages could be retained? The supplemental material may give you some ideas here. You may wish to try multiple classification strategies (but you should at least consider using a naïve Bayes classifier and support vector machine (see package e1071)).

4. Validate or **evaluate** the results of your trained model. Would you use this as a spam filter?

5. Repeat steps 3-4 with various other training/testing pairs. Do you get similar results?