CSCU9YS Assignment

Anna Rasburn (2411187)

# **Milestone 1**

## **Flowchart**

A picture containing screenshot

Description automatically generated

## **Pseudo Code**

1. Get email
2. Prepare email for Data pre-processing
   1. Remove Stop words
   2. Remove Lemmas
   3. Remove punctuation
   4. Remove numbers
3. Create Word Dictionary for each dictionary
   1. Gather words
   2. Find frequency of each word
4. Feature Extraction
   1. Create 30,000 common words per email to compare email body words to.
5. Machine learning
   1. SVM
   2. Naïve Bayes
6. Accuracy
   1. SVM
      1. Precision
      2. Recall
      3. F1
   2. Naïve Bayes
      1. Precision
      2. Recall
      3. F1

Procedure Spam\_Filter{

Body = file.contents()

Data = data\_prep(Body)

function data\_prep(Body){

no\_stopwords = stopwords(body)

no\_punc = remove\_punc(no\_stopwords)

data = lemmatizer(no\_punc)

return data

}

Function create\_dictionary(){

Return dictionary

}

Function feature\_execution(){

Return data

}

Function Machine\_learning(data){

SVM(Data)

Other(Data)

}

Function accuracy(){

}

# **Milestone 2**

## **Text Pre-Processing**

Before you can use the data for classification you must first pre-process the data. This stage consists of three main step.

The first step is to remove stop words from the body of the email. Stop words are words like ‘and’, ‘of’, ‘in’ which have a high frequency in the body of the email. The Stop words have no strong indication of whether an email is spam or not and due to the high frequency of these words, this would skew the classification therefore they should be removed.

The second step is to remove punctuation. Similarly to the stage above, punctuation offers no clear indication of whether an email is spam or not. The high frequency of the punctuation would also skew the clusters in the classification.

The third step is lemmatization which is the process of the getting the a word down to it’s original mean so multiple words can be grouped together into one ‘lemma’ [1]. For example, ‘includes’, ‘included’ and ‘including’ are grouped together into one lemma ‘include’. This step reduces the number of words in our dataset and increases their frequency making them easier to cluster and classify.

## **Machine Learning methods**

### SVM

Naïve Bayes

## **Dataset**

## **Measurement Metric**

# **Milestone 4**

Results

Finding

Discussions

# Reference:

[1]"Stemming and lemmatization", *Nlp.stanford.edu*, 2018. [Online]. Available: https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html. [Accessed: 07- Nov- 2018].

<https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html>