Email Classification

Aim

The aim of this study is to implement text classification of emails using the Naive Bayes classifier for spam detection.

With email dominating as the main form of communication, spam detection to remove spam is important. Email spam is an unsolved global issue that negatively affects productivity and uses up valuable resources. Spam emails are usually sent with malicious intent to users for some monetary gain. Email users spend a large amount of time regularly deleting these spam emails, which needlessly occupy storage space and consumes bandwidth. Hence the development of classifiers that are able to separate legitimate from spam emails is required.

Classification of emails to separate and remove spam from other emails, aims to save the user from having to spend time and effort sorting through and deleting spam, and protect unaware users from malware. Traditional non-machine learning based techniques are easily fooled by spammers and require periodic manual updates. Therefore, machine learning based methods are required, which are able to automatically analyse email contents and dynamically update themselves to cope with new spamming techniques.

Data Preprocessing and Feature Selection

The files in the sample set *Lingpspam-mini600* are represented as a "bag-of-words", where words are extracted from the file and treated as a feature. A feature selection method - *document frequency* - is then used, which picks the best features to use to build a classifier for the given documents. Then the representation is discretised using *tf-idf* to weigh selected words, then normalised using *cosine normalisation*.

1. For each file in the sample set *Lingpspam-mini600*:
    - Open the file and read it line by line.
    - Replace all punctuation and special symbols from each line with a space. (e.g. "don't" becomes "don", "t")
    - The corpus the line belongs to is determined by checking if the line begins with "Subject:". If it does the words within the line will be added to the subject corpus, otherwise they will be added to the body corpus.
    - Words from each line are extracted using space as a delimiter.
    - Stop words, numbers and words containing digits are removed.
    - A counter is created for each word in the email.
    - For each word that appeared in the counter for an email, it is added once to the counter for the corresponding sub-corpus.
2. The words with the top 200 Document Frequency (words which are appeared in the most number of documents) in each corpus was selected for the features of that corpus.
3. For each selected word, its *tf-idf* score is calculated.

4. The *tf-idf* values are normalised using *cosine normalisation*.

The removal of stop words is an important process in text processing, as it filters out extremely common words that have little value in classification. Before stop words were removed, there were 19886 and 1074 unique words in the body and subject corpuses respectively. After their removal, there were 19386 unique words remaining in the body corpus, and 915 words in the subject corpus.

Subject vs Body: Results and Discussion

Corpus: Subject	
	Accuracy [%]
ZeroR	
OneR	
1-NN	
3-NN	
NB	
DT	
MLP	
MyNB	

Corpus: Body	
	Accuracy [%]
ZeroR	
OneR	
1-NN	
3-NN	
NB	
DT	
MLP	
MyNB	

Challenge Results and Discussion

Conclusions

The tokens used in our classifier are formed from single words. Therefore, it will not analyse common consecutive words that are found in spam emails, leading to the failure of our classifier from detecting these emails.

By taking into account permutations of consecutive words, or words that appear within a specified distance of each other, the accuracy of our Bayesian classifier could be increased.

Reflection

Instructions: How to run code