SMS Spam Detection

C.Aditya(201350877),V. HimaSindhuja(201350891)

Objective:

To identify text messages/sms as spam or ham(non-spam).

Challenges:

- Unlike emails, databases for spam SMS are very limited.
- SMSes are limited in length, number of features that can be used for classification is small.
- Text messages generally include abbreviations, informal language, text-speak, other languages written in english.

Dataset:

UCI Machine Learning Repository has a collection of sms messages – SMS Spam Dataset. It contains:

- A collection of 425 spam messages from Grumbletext web site.
- A subset of 3375 ham messages from the NUS Sms Corpus.
- A list of 450 ham messages from Caroline Tag's Phd Thesis.

A total of 4827 ham and 747 spam = 5574 messages
The dataset consists of one message per line. Each line is prefixed with a ham/spam label separated by a tabspace(\t).

Approach / Methodology:

Preprocessing:

- The dataset is split – 70% into training data, 30% to testing data.
- The training data is read line by line and the label and the message are separated.
- The message is tokenized. Only the alphabetical tokens are stored. Tokens with punctuation marks, special characters are ignored. Frequency count of each token per message is stored as Bag of Words.
- No word stemming is done here.
- Three extra attributes per message are also stored – the number of dollar signs, the number of numeric strings, the length of the message.(Message length is a great feature because, spam generaters aim to make of the message length usable without wastage.)
- Now, we remove tokens which occur less than 5 times and more than 500 times. This removes noisy terms and helps normalize the features.
- This phase generates about 1502 + 3 = 1505 features

Feature vectors:

- The feature vectors for each message are computed with their frequency counts as the values. The 3 extra atrributes are also taken as features with their values per each message.

Classification:

- The NaiveBayes classifier is used here.
- Multinomial event model of NB is used as it is useful for feature vectors with word counts.
- Laplace Smoothing is used in NB to handle cases where the feature vectors word counts are zero.

Results:

- The training of the classifier is very fast, < 1Second on a Core i3 2330M with 6GB DDR3 RAM.
- Testing takes far lesser time.
- The accuracy computed by testing the classifier on the 30% test data is 97.43%

Experiments / Observations:

- We trained the classifier on 100% of the training data and tried to classify the smsCorpus_en.xml given to us. It consists of 65,257 messages.
- Some of those messages were repeated in that corpus many times. The classifier labeled 636 messages as spam and 47,345 messages as ham.
- There were false positives(ham labeled as spam) and false negatives(spam labeled as ham). But occurence of false positives was greater than false negatives.

- False positives > False Negatives was because the classifier associated the label 'spam' with features – length of message, numeric terms and word features such as 'free', 'call', 'urgent' etc.
- Presence of false positives and false negatives was because of message content having other language text(hindi, chinese) typed in english and mixed with regular english content. Some of these words are unseen in training data to some extent, but they are handled by laplace smoothing in the training phase to some effect.
- This scenario can get improved accuracy if we introduce some features which use ratio of capital words to small words, ratio of spam words to normal words, consider more different currency symbols, number of URLs in the message.

References:

1) SMS Spam Detection using Machine Learning Approach, Houshmand Shirani-Mehr, hshirani@stanford.edu, "http://cs229.stanford.edu/proj2013/ShiraniMehr-SMSSpamDetectionUsingMachineLearningApproach.pdf"
2) SMS Spam Collection Data Set from UCI Machine Learning Repository, "http://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection"
3) SMS Spam Collection v.1, "http://www.dt.fee.unicamp.br/ ~ tiago/smsspamcollection"