SMS Spam Classification

Raghav Kishan Sunku Ravindranath (820174908), Sathyasagar Nagendra (820859215)
(CS 596 Machine Learning Fall 2016 Final Project Report, San Diego State University)

***Abstract*— Short Message Service (SMS) has an important economic impact for all stakeholders. One important problem in this subject area is spam messages. Many studies have been conducted for the prevention of spam messages. One of the most effective approaches is the Naïve Bayesian algorithm used in filtering techniques. The primary challenge in SMS spam classification is that short messages, in many cases, consist of words composed of abbreviations and idioms. In this project, we design a spam classification technique based on the methods of intelligent spam filtering techniques in the context of mobile text message spam. All methods will not be equally efficient since the characteristics of the SMS content unique. To identify the methods that work best, this project implements some of the popular spam filtering techniques. The technique uses a set of features that will be used as input to the spam classification model. The messages are classified using trained dataset that contains a variety of SMS messages. The results applied to the testing messages show that the proposed system can classify the SMS as spam and ham with high accuracy.**

***Index Terms* —** *Short Message Service (SMS), KNN, Linear Support Vector Machines (SVM), RBF SVM, Decision Tree, Random Forest Classifier, Naive Bayes.*

I. INTRODUCTION

Short messaging service (SMS) has grown to become an indispensable part of modern society. Spammers take advantage of this fact and make use of SMS message to reach potential customers to drive their business interests [1]. This issue is growing by the day, thereby necessitating a mechanism for mobile SMS spam filtering. Mobile SMS spam filtering challenge is similar to email spam filtering, with the difference that they can send a limited number of characters only. It is noticed that almost all spam SMS text may contain a very close pattern due to this limitation. It incorporates some key words to attract potential customers and then some contact information, usually a call back number, reply SMS number or a Uniform Resource Locator (URL) that they can visit [2].

The fact that the number of characters in each message is limited should make it possible for the search methods to come out with better results. The spam filtering problem essentially is a case of text classification [1]. This project implements and evaluates the popular algorithms used for spam classification on publicly available SMS spam corpus to identify the better methods so that they can be further optimized for the SMS text paradigm. The goal is to apply few of the popular machine learning algorithms regarding SMS spam classification problem, compare their performances, and further explore the problem. We further plan to design an application based on one of these algorithms that can filter SMS spams with high accuracy.

This paper is organized as follows. Section II discusses the task description indicating the major steps involved in the project. Section III presents some of the major challenges involved in SMS span message classification. Section IV provides information about the dataset collected for the experiments, evaluation metrics used, major empirical results, analysis leading to the empirical results. Section V concludes the report and describes the prospective future work.

II. Task Description

We tackle the problem of SMS spam classification with the below seven step process.

(1) Prepare the corpus - A corpus is a large, structured set of texts electronically stored and processed used to perform statistical analysis and hypothesis testing, checking occurrences or validating rules within a specific language territory. This project uses corpus which has been collected from free or free for research sources available on the Internet.
(2) Identify real words – Text messages may contain words like “hi”, “HI”, “Hi!!”. All these words must be considered as the same.
(3) Clean the corpus – From the corpus, non-content words like “me”, “myself”, “ours” and excess spaces are removed.
(4) Use a classifier algorithm to detect spam or ham – In this project we implement the following algorithms. KNN, Linear SVM, RBF SVM, Decision Tree, Random Forest Classifier and Naive Bayes.
(5) Divide corpus into training and testing data.
(6) Apply the model on training and test data.
(7) Analyze the results.

III. Major Challenges

(1) Databases for SMS spams are very limited.
(2) Due to the small length of text messages, the number of features that can be used for classification is small.
(3) A header does not exist for SMS messages.
(4) Text messages are full of abbreviations and have much less formal language.
(5) High false positive rate is a major challenge with bulk sending using behavioral based detection.

IV. Experiments

A. Dataset Description [6]

We use the public set of SMS labeled messages - SMS Spam Collection v.1 - that have been collected for mobile phone spam research. It has one collection composed by 5,574 English, real and non-encoded messages, tagged according being legitimate (ham) or spam. This corpus has been collected from free or free for research sources at the Internet.

A collection of 425 SMS spam messages was manually extracted from the Grumbletext Web site. This is a UK forum in which cell phone users make public claims about SMS spam messages, most of them without reporting the very spam message received. The identification of the text of spam messages in the claims is a very hard and time-consuming task, and it involved carefully scanning hundreds of web pages. The Grumbletext Web site is: http://www.grumbletext.co.uk/.

A subset of 3,375 SMS randomly chosen ham messages of the NUS SMS Corpus (NSC), which is a dataset of about 10,000 legitimate messages collected for research at the Department of Computer Science at the National University of Singapore. The messages largely originate from Singaporeans and mostly from students attending the University. These messages were collected from volunteers who were made aware that their contributions were going to be made publicly available. The NUS SMS Corpus is avalaibleat http://www.comp.nus.edu.sg/~rpnlpir/downloads/corpora/smsCorpus/.

A list of 450 SMS ham messages collected from Caroline Tag's PhD Thesis available at http://etheses.bham.ac.uk/253/1/Tagg09PhD.pdf. Finally, we have incorporated the SMS Spam Corpus v.0.1 Big. It has 1,002 SMS ham messages and 322 spam messages and it is public available at: http://www.esp.uem.es/jmgomez/smsspamcorpus/.

B. Evaluation Metrics

To visualize the performance of an algorithm, typically a supervised learning a confusion matrix is used. Also, known as error matrix, each column of the confusion matrix signifies an instance of a predicted class and each row signifies an instance of the actual class.

		Prediction	
		$\hat{y}=1$	$\hat{y}=0$
Groundtruth	$y=1$	True-positive	False- Negative
	$y=0$	False-positive	True-negative

The above table is an example of a confusion matrix. If the prediction and ground truth are equal, then it is either True-positive or True negative based on the classification labels. If the prediction is not equal to the ground truth, then it is either False-positive or False-Negative based on the classification labels.

Our computed values of the confusion matrix for the Naïve Bayes algorithm is as follows:

	Prediction		
Ground Truth		Spam	ham
	Spam	91	59
	ham	0	965

It can be observed from the above table that, the Naïve Bayes algorithm has 0 for False-Positive, 59 for False-Negative, 91 for True-Positive and 965 for True-Negative. From this we can derive that the classification has occurred fairly accurate.

For the SVM algorithm, the confusion matrix is as follows:

	Prediction		
Ground Truth		Spam	ham
	Spam	133	17
	ham	3	962

It can be observed from the above table that, the SVM algorithm has 3 for False-Positive, 17 for False-Negative, 133 for True-Positive and 962 for True-Negative. From this we can derive that the classification has occurred fairly accurate.

From these confusion matrices, the accuracy, precision, and recall rates can be determined. These values are the used to identify the perfect algorithm for the given data set and classification problem.

C. Major Results

As indicated above, for visualizing the results we use confusion matrix. From the confusion matrix, Accuracy, Precision and Recall can be determined which enables us to determine the correct algorithm for the task of SMS classification.

Accuracy is the percentage of correct classification of the input data. It can be determined using the following formula:

$$\text{Accuracy} = \frac{|\{\text{True-positive}\} + \{\text{True-Negative}\}|}{|\{\text{Positive}\} + \{\text{Negative}\}|}$$

Precision is the fraction of the documents retrieved that are relevant to the user's information need.

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

In binary classification, positive predictive values are analogous to precision. Precision considers all retrieved documents. This measure is called 'P@n'.

Recall is the portion of documents are relevant to the query that are successfully retrieved.

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

Sensitivity is the other name given to recall in binary classification. It is understood as the probability that a query retrieves a relevant document.

Fall-out is the portion that is retrieved which is non-relevant. It can be calculated as follows.

$$\text{fall-out} = \frac{|\{\text{non-relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{non-relevant documents}\}|}$$

Fall-out is closely related to specificity in binary classification and is equal to (1 - specificity). It can be understood as the probability that a non-relevant document is retrieved.

In the project, for the Naïve Bayes algorithm, the Accuracy, precision, recall and fall out values are as follows.

Accuracy = 94.70 %

	precision	recall	f1-score
ham	0.94	1	0.97
spam	1	0.61	0.76
avg/total	0.95	0.95	0.94

For SVM, the Accuracy, precision, recall and fall out values are as follows.

Accuracy = 98.20 %

	precision	recall	f1-score
ham	0.98	1	0.99
spam	0.98	0.89	0.93
avg/total	0.98	0.98	0.98

D. Analysis

In this project, we used 6 different algorithms to identify the perfect algorithm to classify the SMS into ‘spam’ and ‘ham’. As indicated before the data corpus has been divided into ‘train’ and ‘test’ data. The test data has been applied on these algorithms such that, ten different portions of the data are selected randomly. Nine portions are used for training the model and one portion for validation. The six algorithms that were used are 'Linear SVM', 'RBF SVM', 'Decision Tree', 'KNN', 'Random Forest Classifier', and, 'Naive Bayes'. Thus, six set of results were obtained. A brief description of the algorithms is as follows:

Support Vector Machine – SVM

SVM is a machine learning algorithm which is a supervised learning model used for classification. The corpus data is represented as a set of points that are mapped so that points of separate categories are divided by a clear gap that is as wide as possible. SVM is divided into two, Linear SVM and RBF SVM.

Decision Trees

Decision trees learning is one in which a decision tree is used as a predictive model which maps observations of an object to the conclusions of the target object. When the target variables in a tree model is a finite set, then it is called a classification tree. The leaves of the tree represent the labels and branches denote conjunctions of features that lead to class labels.

K- Nearest Neighbors - KNN

The KNN algorithm is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether KNN is used for classification or regression.

Random Forest Classifier

Random forests or random decision forests are an ensemble learning method for tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the made of the classes (classification) or mean prediction (regression) of the individual trees.

Naive Bayes

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong independence assumptions between the features. Naive Bayes classifiers are highly scalable and requires several parameters linear in the number of variables in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by iterative approximation as used for many other types of classifiers.

The results obtained from the 6 algorithms are as follows:

Model	SD	Accuracy %
KNN	0.0042	92.26%
Linear SVM	0.0009	86.61%
RBF SVM	0.0061	96.17%
Decision Tree	0.0088	94.44%
RandomForestClassifier	0.0009	86.61%
Naive Bayes	0.0027	96.10%

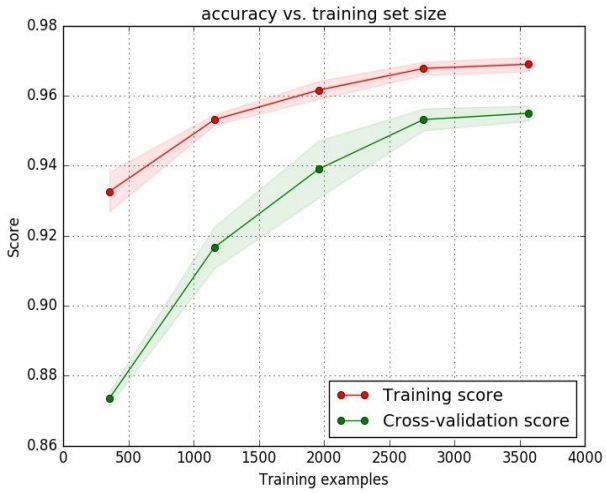
From the above results, we decided to select two algorithms on which we applied the test corpus.

1. Naive Bayes
2. SVM

From the Results and Evaluation Metrics sections, it can be observed that Naïve Bayes algorithm has an

accuracy is 94.70 % and SVM – Support Vector Machine has an accuracy of 98.20 %. These results are quite accurate which enables us to use it in real time applications.

Plot:



The above plot indicates the training score and cross validation score w.r.t size of the training corpus. The accuracy of the result increases with increase in size of the corpus.

V. Conclusion and Future Works

The accuracy findings of the classification models applied to the SMS Spam dataset are presented in table IV. From the results, it is evident that Naive Bayes and RBF SVM are among the best algorithms for SMS spam detection. The best classifier in the original paper citing this dataset is the one utilizing RBF SVM as the learning algorithm, which yields overall accuracy of 98.20%. The second-best classifier is the Naive Bayes classifier with overall accuracy of 94.70%. The overall error is reduced by more than half when compared to the result of previous work. This is achieved since meaningful features such as the length of messages in number of characters, certain thresholds for the length, learning curves and misclassified data have been considered. In the future, we plan to perform thorough experiments with machine learning content based classifiers to improve the results of previous work by us and others on the much smaller SMS Spam Corpus.

References

[1] Paul Graham, (August 2002), A plan for spam, viewed: 28 September 2011, http://paulgraham.com/spam.html

[2] Duan, L., Li, N., & Huang, L. (2009). "A new spam short message classification" 2009 First International Workshop on Education Technology and Computer Science, 168-171.

[3] GÃ³mez Hidalgo, J.M., Cajigas Bringas, G., Puertas Sanz, E., Carrero GarcÃa, F. Content Based SMS Spam Filtering. Proceedings of the 2006 ACM Symposium on Document Engineering (ACM DOCENG'06), Amsterdam, The Netherlands, 10-13, 2006.

[4] Cormack, G. V., GÃ³mez Hidalgo, J. M., and Puertas SÃ¡nz, E. Feature engineering for mobile (SMS) spam filtering. Proceedings of the 30th Annual international ACM Conference on Research and Development in information Retrieval (ACM SIGIR'07), New York, NY, 871-872, 2007.

[5] Cormack, G. V., GÃ³mez Hidalgo, J. M., and Puertas SÃ¡nz, E. Spam filtering for short messages. Proceedings of the 16th ACM Conference on Information and Knowledge Management (ACM CIKM'07). Lisbon, Portugal, 313-320, 2007.

[6] Dataset: http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/s