2016 12th World Congress on Intelligent Control and Automation (WCICA)
June 12-15, 2016, Guilin, China

A Method of SMS Spam Filtering Based on AdaBoost Algorithm

Xipeng Zhang, Gang Xiong*, Yuexiang Hu, Fenghua Zhu, Xisong Dong, Timo R. Nyberg

***Abstract*—Short message is one of the most common communication media for mobile subscribers, so major mobile operators are devoted to improve their Short Message Service (SMS). However, the annoying and undesired messages, also named message spam or simply spam, not only worsen the users' experience, but also cause their complaints on SMS. In this paper, we present a novel Chinese SMS spam filtering framework based on AdaBoost algorithm to provide accurate and effective short messages classification. Three content-based weak filters are introduced to boost the performance of final classification decision. Results from Receiver Operating Characteristics (ROC) analysis prove the proposed method has such advantages as higher efficiency and fewer parameters over those established SMS spam filtering methods. The application of the proposed method is expected to block the most spam for mobile subscribers and improve the service quality of SMS. With simple data processing and few training parameters, the proposed method can be applied into the practice of short text classification.**

I. Introduction

Along with the rapid development of mobile communication technology, short message has become a popular way for mobile subscribers to send message since the 21th century. In China, between 2005 and 2010, the text message volume per year has grown exponentially (304.65 billion to 831.7 billion), which greatly changed the communication way. Additionally, as a mobile value-added service, Short Message Service (SMS) is enriched and improved by all major mobile operators both in coverage and transmission rate. Unfortunately, huge amount of unsolicited junk messages flood users' inboxes, which leads to valuable time lost, bandwidth cost and lots of unnecessary trouble [1]. These annoying and undesired messages, also named message spam or simply spam, not only worsen user experience, but also cause complaints on SMS. What's worse, the ease of generating content and decreasing cost of sending spam messages make the problem more and more serious. In recent years, governments worldwide have shown increasing concern on the issue of spam abuse and taken legal, economical and technical measures to solve the problem.

Given the long-term of legislation and lack of economic penalties, technical measures against spam can be an available and effective way to ease the problem. A spam filter is designed to block the most spam messages and minimize their adverse impact. Since there are some similarity between junk e-mail and spam message, some proposed approaches to deal with junk e-mails may contribute to the design of SMS spam filter. For example, in each e-mail, content-based filtering, the most popular method focuses on spam features such as indicative words (e.g., "free", "Viagra", etc.), unusual distribution of punctuation marks and capital letters (e.g., "BUY!!!!!!"), etc [2]. And the use of blacklist marks a list of e-mail addresses which are known belong to spammers. Compared with e-mail, SMS message is shorter and lack hypertext structure, so not all existing techniques used to detect junk e-mails can be easily transferred to the problem of mobile spam. Meanwhile, the absence of message spam database and insufficient information exchange and feedback between subscribers and operators make it more challenging to find a simple but effective spam filter.

The classifier design is the core of building a good spam filter. In this paper, we propose a SMS spam filtering method to provide accurate and effective recognition of spam and ham (non-spam). There is no explicit restriction on feature selection and parameter estimation of each weak learner, which allows possible extension of the method. The goal of boosting is to produce single strong classifier through the combination of these weak learners, which has much better performance than monolithic learner [3]. Experiments on data set from 5-fold cross-validation suggest that SMS spam filtering with AdaBoost algorithm outperforms these established SMS spam filtering methods. The advantages of the proposed method are outlined as follows:

- Since the component classifiers only need to perform better than chance, there is no explicit requirement on data processing, which will save time and provide flexibility.
- With the application of increment algorithm, the model can be updated easily when new data comes. The evaluation results show the improved accuracy and low generalization error compared to classical SMS spam filtering approaches, such as Naive Bayes (NB) [4-6], C4.5 [7], Support Vector Machines (SVM) [8].

This work was supported in part by the National Natural Science Foundation of China under Grants 71232006, 61233001, 61304201 and 61174172; Finnish TEKES's project"SoMa2020: Social Manufacturing" (2015-2017); Chinese Guangdong's S&T project (2014B010118001, 2014B090902001, 2014A050503004), and Chinese Dongguan's Innovation Talents Project (Gang Xiong).

X. P. Zhang is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation of Chinese Academy of Science, Beijing, CO 100190 China (email: zhangxipeng15@mails.ucas.ac.cn).

G. Xiong is with the Cloud Computing Center, Chinese Academy of Sciences, Dongguan, CO 523000, China (*Correspondence author e-mail: gang.xiong@ia.ac.cn).

Y. X. Hu is with Hainan Zhongke Flower Ocean Cloud Commerce Technology Co. Ltd, Haikou, CO570311, Hainan, China.

F. H. Zhu is with the Beijing Engineering Research Center of Intelligent Systems and Technology, Institute of Automation, Chinese Academy of Sciences, Beijing, CO 100190 China.

X. S. Dong is with the Qingdao Academy of Intelligent Industries, Qingdao, CO 266061, China.

T. R. Nyberg is with Department of Industrial Engineering and Management, Aalto University, Aalto, CO FI-00076 Finland.

• Due to its simple classification model and few training parameters, the technique discussed in this paper can be easily extended and flexibly adopted in web short text classification.

The paper is organized as follows. The introduction of the problem and some issues about data processing are presented in section II. The proposed design method of several weak classifiers and AdaBoost algorithm are explained in section III. The performance evaluation of the proposed method with the help of cross validation is discussed in Section IV. Finally, the paper is concluded in section V, which presents conclusion, commercialization of the research achievement and future scope for the proposed filter.

II. Data Processing

The selection and evaluation of classifier is driven by what kind of data we collect. One classifier may have exactly opposite effects on different data sets. Before our design of SMS spam filter, we must pay close attention to the collection of messages. We have built a collection of 800,000 messages in Chinese. Each message can be interpreted as a text document, which contains only character strings. And, in front of each message, there is a class identifier 0 or 1, labeling as ham or spam respectively. After simple statistics, we have got to know that there are 720,000 (90%) legitimate messages and 80,000 (10%) spam messages, which means a trivial rejecter always in favor of the majority class (ham, in our case) would show an accuracy of 0.9.

A. Message Pre-processing

A message may contain characters, digits, punctuations, and symbols etc., which needs to be processed and qualified. Message pre-processing is the most important step before feature selection and classifier design. A bad representation of collection data may lead to low efficiency and poor accuracy of classifier. Traditional text processing includes the following steps:

1) Stop-word removal: stop-words are frequently occurring and insignificant words in a language that help construct sentences but do not represent any content of the documents. Such words should be removed before messages are indexed and stored.
2) Words Segmentation: the goal of word segmentation is to divide a sequence of characters into several meaningful words. Chinese lexical analysis system, developed by Institute of Computing Technology, Chinese Academy of Sciences can achieve the overall accuracy of 97.58%, with a speed of 35.1KB/s.
3) Low-frequency words filtering: words whose occurrences are less than a threshold frequency can be removed to reduce the amount of words for each query. The threshold frequency can be gotten from the analysis of word frequency.
4) Others tasks: character conversion, synonym substitution and digits removal (expect some specific types).

B. Feature Selection

After message pre-processing, almost 80,000 words are derived from the data set. In practice, we hope to find a small but significant attributes so that the input data is stripped of unnecessary feature as much as possible. Since feature selection for text classification has been a noteworthy problem for more than a decade, many attribute quality criteria have been presented in the literature. Researches show that χ^2 statistics and Information Gain (IG) are both computationally scalable and high-performing to large collections [9].

We randomly choose some samples to compare the performance of these two feature selection methods using four classifiers (NB, Rocchio, *k*NN and SVM). Experiments suggest that χ^2 is more suitable in this case.

III. The Design of SMS Message Classifier

In this section, we mainly describe the principle of the design of weak classifiers themselves and the application of AdaBoost algorithm on the problem. Taking filter efficiency and classification accuracy into consideration, three simple weak classifiers based on different pattern features are introduced. The final classifier combines these component classifiers together to form an ensemble whose joint decision has arbitrarily high accuracy.

In order to avoid confusion and eliminate ambiguity, we make the following agreements in the description of classifier design:

• We let c donate the state of nature, with $c=c_1$ for spam and $c=c_2$ for ham.

• The output given by each component classifier is -1 or +1, which represents the classification decision for responding test massage is ham or spam respectively.

• The collected data sat $\mathcal{D}$ is randomly spilt in to two parts. The first part (e.g., 90% of the patterns) is used as a standard training set for setting free parameter in each classifier model; the other (e.g., 10%) is the validation set and is meant to represent the full generalization task.

• 5-fold cross-validation method is used for training or parameter adjustment (if there is any) in the design of weak classifier. Since our ultimate goal is to minimize generalization error for test set, training or parameter adjustment in cross validation should be stopped at the first minimum of the validation error, as sketched in Fig. 1[10].

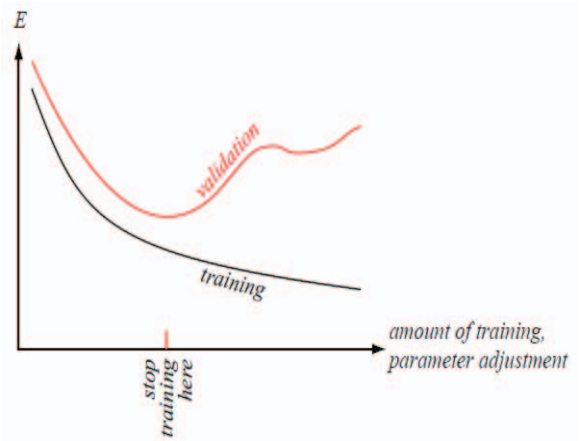

Fig. 1. Generalization error vs. amount of training or parameter adjustment

A. Bayesian decision making based on minimum risk

Statistics results show that the most spam messages are aimed at particular group for the purposes such as harassment, advertising and financial fraud. Spammers tend to send much longer messages than normal people to make them more attractive and deceptive. Table I presents the statistics related to message length extracted from training data, which show consistency with our prior knowledge. Message length difference between spam and ham suggests that message length may be a useful feature to explore in our classifier design.

Table Ⅰ: Message length difference between spam and ham

Statistics \ Class	Spam	Ham
Mean	66.37	21.14
Variance	1121.2	92.5

Bayesian decision theory is a fundamental statistical approach based on the quantifying the tradeoff between various classification decisions and the costs that accompany with such decisions [10]. Since the ratio of spam to ham is 1:9, we can reasonably assume that the prior probability of spam $P(c_1)$ is 0.1 and that of ham $P(c_2)$ is 0.9. Sorted frequency distribution of message length for each class can approximately represent the actual class-condition probability density function $p(x|c)$ in the design of weak learner. As shown in Fig. 2, two curves describe the difference in message length between population of spam and ham.

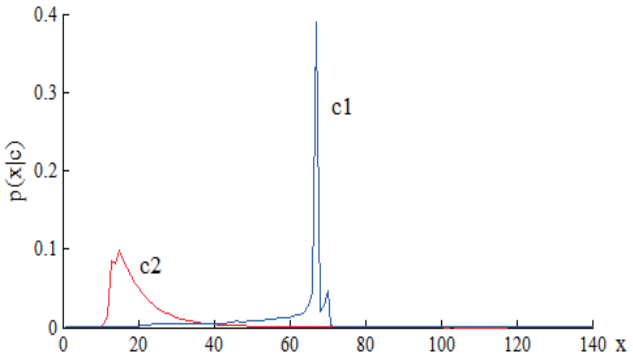

Fig. 2. Approximate class-conditional probability density function of message length

Given different cost when some kind of mistakes happen, lost function λ_{ij} shows the loss incurred for deciding c_i when the true state of nature is c_j, as outlined in Table II, where r is a free parameter. Ordinarily, the loss incurred for mistaking a ham is greater than the loss incurred for mistaking a spam, which indicates $r > 1$.

The Bayesian decision rule to minimize conditional risk is given as follows:

$$\text{Decide } c_1 \text{ if } \frac{p(x \mid c_1)}{p(x \mid c_2)} > \frac{\lambda_{12}}{\lambda_{21}} \frac{P(c_2)}{P(c_1)} = 9r \text{ ; otherwise decide } c_2$$

The resulting minimum Bayes risk is the best performance than can be achieved on certain pattern feature.

Table Ⅱ: Loss function

λ Actual \ Predicted	Spam	Ham
Spam	0	1
Ham	r	0

B. Naive Bayes

SMS spam filtering is some kind of a text classification or categorization problem, whose objective is automatically to classify future document on the basis of classification model from labeled documents. Naive Bayes is the most immediate and effective Bayesian learning method for text classification. Therefore, the naive Bayes can be a good choice to construct a weak SMS message classifier.

The naive Bayes for text is derived based on a probabilistic generative model, which treats each short message as a bag of words. Formally, let $D=\{D_1, D_2\}$, where D_j is the subset of messages for class c_j, j=1, 2. The vocabulary $V=\{w_1, w_2, \ldots, w_{|V|}\}$ is the set of all distinction words in D, $|V|$ being the number of words in V. Given a test example $\boldsymbol{d}$, with observed attribute words $w_{\boldsymbol{d},1}$ through $w_{\boldsymbol{d},|\boldsymbol{d}|}$, where $w_{\boldsymbol{d},k}$ is the word in position k of message $\boldsymbol{d}$ and $|\boldsymbol{d}|$ is the length of d, k=1, 2, …, $|\boldsymbol{d}|$. The Bayesian prediction for d is the class c_j with the highest posterior probability $P(c_j \mid w_{\boldsymbol{d},1}, w_{\boldsymbol{d},2}, \ldots, w_{\boldsymbol{d},|\boldsymbol{d}|})$, which can be expressed as:

$$\begin{aligned} P(c_j \mid w_{\boldsymbol{d},1}, w_{\boldsymbol{d},2}, \ldots, w_{\boldsymbol{d},|\boldsymbol{d}|}) &= \frac{P(w_{\boldsymbol{d},1}, w_{\boldsymbol{d},1}, \ldots, w_{\boldsymbol{d},|\boldsymbol{d}|} \mid c_j) P(c_j)}{P(w_{\boldsymbol{d},1}, w_{\boldsymbol{d},1}, \ldots, w_{\boldsymbol{d},|V|})} \\ &= \frac{P(w_{\boldsymbol{d},1}, w_{\boldsymbol{d},1}, \ldots, w_{\boldsymbol{d},|\boldsymbol{d}|} \mid c_j) P(c_j)}{\sum_{j=1}^{2} P(w_{\boldsymbol{d},1}, w_{\boldsymbol{d},1}, \ldots, w_{\boldsymbol{d},|\boldsymbol{d}|} \mid c_j) P(c_j)} \end{aligned} \quad (1)$$

$P(c_j)$ is the class prior probability of class c_j, which can be estimated from the training data.

Specifically, the generative model makes the assumption that words are conditionally independent given class label. We then obtain:

$$P(w_{\boldsymbol{d},1}, w_{\boldsymbol{d},2}, \ldots, w_{\boldsymbol{d},|\boldsymbol{d}|} \mid c_j) = \prod_{k=1}^{|\boldsymbol{d}|} P(w_{\boldsymbol{d},k} \mid c_j) \quad (2)$$

Experiment comparisons show that multinomial formulation consistently produces more accuracy classifiers than multi-variable Bernoulli formulation, which indicates that term frequency need to be considered [11]. Thus, the estimated conditional probability $P(w_{\boldsymbol{d},k} \mid c_j)$ is simply the number of times that $w_{\boldsymbol{d},k}$ occurs in the training data D_j (of class c_j) divided by the total number of word occurrences in the training data for that class:

$$P(w_{\boldsymbol{d},k} \mid c_j) = \frac{\sum_{i=1}^{|D_j|} N_{d,k,i}}{\sum_{s=1}^{|V|} \sum_{i=1}^{|D_j|} N_{si}} \quad (3)$$

Lidstone's law of succession is a kind of smoothing method to handle 0 counts for infrequently occurring words that do not appear in the training set. The revised conditional probability $P(w_{\boldsymbol{d},k} \mid c_j)$ makes sure any word will have at least a very small probability of occurrence, as shown below:

$$P(w_{\boldsymbol{d},k} \mid c_j) = \frac{\lambda + \sum_{i=1}^{|D_j|} N_{\boldsymbol{d},k,i}}{\lambda |V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D_j|} N_{si}} \quad (4)$$

where λ is a fraction of a word in both the numerator and the denominator. Experiments through cross-validation show that optimal λ value for the validation set is 0.76.

Substituting Equations (2), (4) into the primal posterior (1), by Bayes rule, we have the most probable class for the test message $\boldsymbol{d}$:

$$c = \arg\max_{c_j} P(\mathrm{c}_j) \prod_{k=1}^{|\boldsymbol{d}|} P(w_{\boldsymbol{d},k} \mid \mathrm{c}_j) \quad (5)$$

With the application of increment algorithm, the model can be updated easily as new data comes.

C. k-Nearest Neighbor

Spam samples have some similarity with each other, which may differ from that of ham samples. Based on this assumption, the idea of *k*-nearest neighbor (*k*NN) can be found use in short message classification. Given a test instance $\boldsymbol{d}$, the decision rule of *k*NN is taking the most frequent class among the *k* nearest neighbor of $\boldsymbol{d}$. The general *k*NN algorithm is given as below:

Algorithm *k*NN(*D*, *d*, *k*)

1 Compute the distance or similarity between $\boldsymbol{d}$ and every example in *D*
2 Choose the *k* examples in *D* that are nearest to *d*, donate the set by P
3 Assign *d* the class that is the most frequent (or the majority) class in P

The core of *k*NN algorithm is the definition of distance or similarity function. For text categorization, cosine similarity is the most well known similarity measure. The similarity between test sample $\boldsymbol{d}$ and training sample $\boldsymbol{m}_j$ is

$$sim(\boldsymbol{m}_j, \boldsymbol{d}) = \frac{\langle \boldsymbol{m}_j \cdot \boldsymbol{d} \rangle}{\|\boldsymbol{m}_j\| \cdot \|\boldsymbol{d}\|} = \frac{\sum_{i=1}^{|V|} \alpha_{ij} \cdot \alpha_{id}}{\sqrt{\sum_{i=1}^{|V|} \alpha_{ij}^2} \cdot \sqrt{\sum_{i=1}^{|V|} \alpha_{id}^2}} \quad (6)$$

where α_{ij} and α_{id} is the term frequency-inverse document frequency (TF-IDF) of w_i (a term) in $\boldsymbol{d}$ and $\boldsymbol{m}_j$ respectively and $|V|$ is as stated early. The available TF-IDF term weight α_{ij} is the product of normalized term frequency (donated by tf_{ij}) and inverse document frequency (donated by idf_i), which are given by

$$\alpha_{ij} = tf_{ij} \cdot idf_i \quad (7)$$

$$tf_{ij} = \frac{f_{ij}}{\max\{f_{1j}, f_{2j}, ..., f_{|V|j}\}} \quad (8)$$

$$idf_i = \log \frac{N}{df_i} \quad (9)$$

Additional notations are as follows:

f_{ij} : the row frequency count of term w_i in message $\boldsymbol{m}_i$

N : the total number of messages in the collection

df_i : the number of messages that contain the term w_i

A range of *k* values have been tried and when *k*=3, the approach reaches the balance between accuracy and efficiency. *k*NN is a simple and flexible learning method with good performance as those elaborated methods.

D. AdaBoost Algorithm

Boosting is a general learning algorithm, which has proven effective when used in conjunction with any of a wide range of basic classifier methods. The most popular, AdaBoost—from adaptive boosting—combines a number of weak learners to form a ensemble whose joint performance has been boosted [10]. The following pseudo code shows the execution of the AdaBoost algorithm.

Algorithm AdaBoost(*D*, k_{max})

1 **begin initialize** D=$\{\boldsymbol{m}_1, y_1; \boldsymbol{m}_2, y_2; \ldots; \boldsymbol{m}_n, y_n\}$, $W_1(i)=1/n$, k_{max}
2 $k \leftarrow 0$
3 **do** $k \leftarrow k+1$
4 Train weak learner C_k using D sampled according to distribution $W_k(i)$
5 $E_k \leftarrow$ Training error of C_k measured on D using $W_k(i)$
6 $\alpha_k \leftarrow \frac{1}{2} \ln \frac{1-E_k}{E_k}$
7 $W_{k+1}(i) \leftarrow \frac{W_k(i)}{Z_k} \times \begin{cases} e^{-\alpha_k} & \text{if } h_k(\boldsymbol{m}_i)=y_i \text{ (correctly classified)} \\ e^{\alpha_k} & \text{if } h_k(\boldsymbol{m}_i) \neq y_i \text{ (incorrectly classified)} \end{cases}$
8 **until** $k=k_{max}$
9 **return** $\boldsymbol{C}_k$ and α_k for $k=1$ *to* k_{max} (ensemble of classifiers with weights)
10 **end**

The final discriminant function is merely weighted voting of the outputs given by the component classifiers, as shown below.

$$g(\boldsymbol{m}) = \mathrm{sgn}\left[\sum_{k=1}^{k_{\max}} \alpha_k h_k(\boldsymbol{m})\right] \quad (10)$$

Proofs have shown that the training error drops exponentially fast with the number of component classifiers. In order to avoid overfitting, the selection of k_{max} is the tradeoff between classification accuracy and generalization error, in oue case, k_{max}=3.

IV. Evaluation

In SMS spam filtering, we are typically interested in only the minority spam class. Moreover, with the high proportion of ham in the collection, accuracy is a suitable evaluation method in such a case. The Receiver Operating Characteristics (ROC) analysis is the most suitable evaluation method for such an imprecise environment [11]. Instead of single value of accuracy, a pair of values is introduced to compare the performance of different classifiers. The referred values are True Positive Rate (TPR) and False Positive Rate (FPR), which can be easily explained through a confusion matrix (Table III).

Table III Confusion matrix of a classifier

Actual \ Predicted	Classified Positive	Classified Negative
Actual Positive	True Positive	False Negative
Actual Negative	False Positive	True Negative

$$TPR = \frac{TP}{TP + FN} \quad (11)$$

$$FPR = \frac{TP}{TP + FP} \tag{12}$$

The TPR is basically the recall of the positive and the FPR is equally 1 less the recall of the recall of the negative [2]. Actually, a ROC curve is a plot of the TPR against the FPR.

Based on established method on spam filtering, the following two filters are selected for comparison:

- SVM: a linear learning system with high accuracy for text classification.
- C4.5: a simple and effective learning method which output a decision tree.

We have evaluated these filters on the collection $\mathcal{D}$ with 5-fold cross-valuation. Classification results in form of ROC curve for different classifiers are shown in Fig. 3.

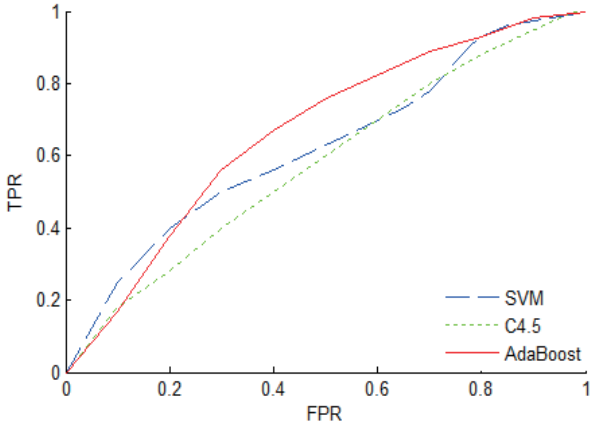

Fig. 3. ROC curves of different classifiers

The area under the ROC curve (AUC) is an overall performance evaluation criterion for ranking the different methods. From Fig. 3, it would be realized that AdaBoost performs over other algorithms in most case. Furthermore, due to its simple classification model and small amount of online calculation, the proposed method is expected to achieve better performance and increased efficiency in practice. We can conclude that ensemble of weak learners with AdaBoost algorithm tends to provide accurate SMS spam filtering and accordingly ease the problem of spam abuse.

V. Conclusion and Future Work

With the integration of 3 weak learners, the final classifier with AdaBoost algorithm is able to ensure high accuracy of SMS spam filtering. Data processing and model selection are based on the tradeoff between generalization errors and filter efficiency. Results from ROC analysis suggest the proposed method have advantages of high efficiency and few parameters over some established methods.

Now, we are working on optimal feature selection and possible extension of the method. The application of the method is expected to block most spam for mobile subscribers and improve the quality of SMS.

References

[1] Pera M.S., and Ng Y.K., "Using word similarity to eradicate junk emails", *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, Lisboa, Portugal, Nov. 6-8, 2007, pp. 943-945.

[2] Hidalgo J. M. G., Bringas G. C., Sánz E. P., "Content based SMS spam filtering", *Proceedings of 2006 ACM Symposium on Document engineering*, Amsterdam, Netherlands, Oct. 10-13, 2006, pp. 107-114.

[3] Zaidi N.A., Suter D., "Confidence rated boosting algorithm for generic object detection", 19th *International Conference on Pattern Recognation (ICPR)*, Tampa, Florida, USA, Dec. 8-11, 2008, pp. 1-4.

[4] Agarwal S., Kaur S., Garhwal S., "Antispammer for mobile messages", *Proceedings of the 6th International Conference on Computer and Communication Technology*, Allahabad, India, Sept. 25 - 27, 2015, pp. 26-30.

[5] Ahmed I., Guan D.C., Chung T., "Anovel semi-supervised learning for SMS classfication" *Proceedings of the 2014 International Conference on Machine Learning and Cybernetics*, Lanzhou, China, Jul. 13-16, 2014, pp.856-861.

[6] Agarwal A., Gupta B., Bhatt G., Mittal A., "Construction of a semi-automated model for FAQ retrieval via ahort message service", *Proceedings of the 7th Forum for Information Retrieval Evaluation*, Gandhinagar, India, Dec. 4-6, 2015, pp. 35-38.

[7] Qaroush A., Khater I.M., Washaha M., "Indentifying spam e-mail based-on statistical header features and sender behavior", *Proceedings of the CUBE International Information Technology Conference*, Pune, Maharashtra, India, Sept. 3-5, pp. 771-778.

[8] Narayan A., Saxena P., "The curse of 140 characters: evaluating the efficacy of SMS spam detection on Android", *Proceedings of the 3rd ACM workshop on Security and privacy in smart phones & mobile devices*, Berlin, Germany, Nov. 8, 2013, pp.33-40.

[9] Rogati M., Yang Y.M., "High-performing feature selection for text classification", *Proceedings of the eleventh international conference on Information and knowledge management*, McLean, Virginia, USA, Nov. 4-9, 2002, pp. 659-661.

[10] Stock D.G., Duda R.O., Hart P.F., *Pattern Classification* (*Second edition*), and chapter 9, pp. 24-28.

[11] Liu Bing, *Web Data Mining* (*Second edition*) chapter 3, pp. 82-85.