Mobile Junk Message Filter Reflecting User Preference

Kyoung-Ju Lee[1,2] and Deok-Jai Choi[1]
[1] School of Electronics and Computer Engineering, Chonnam National University
Gwangju – South Korea
[2] Department of Global and Enterprise Business , Korea Telecommunications
Seongnam Gyeonggido – South Korea
[e-mail: 106727@ejnu.net, dchoi@jnu.ac.kr]
*Corresponding author: : KyoungJu Lee

Received April 9, 2012; revised July 6, 2012; accepted August 16, 2012;
published November 30, 2012

Abstract

In order to block mobile junk messages automatically, many studies on spam filters have applied machine learning algorithms. Most previous research focused only on the accuracy rate of spam filters from the view point of the algorithm used, not on individual user's preferences. In terms of individual taste, the spam filters implemented on a mobile device have the advantage over spam filters on a network node, because it deals with only incoming messages on the users' phone and generates no additional traffic during the filtering process. However, a spam filter on a mobile phone has to consider the consumption of resources, because energy, memory and computing ability are limited. Moreover, as time passes an increasing number of feature words are likely to exhaust mobile resources. In this paper we propose a spam filter model distributed between a users' computer and smart phone. We expect the model to follow personal decision boundaries and use the uniform resources of smart phones. An authorized user's computer takes on the more complex and time consuming jobs, such as feature selection and training, while the smart phone performs only the minimum amount of work for filtering and utilizes the results of the information calculated on the desktop. Our experiments show that the accuracy of our method is more than 95% with Naïve Bayes and Support Vector Machine, and our model that uses uniform memory does not affect other applications that run on the smart phone.

Keywords: SMS spam filter, smart phone application, personalized spam filter

"This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency)" (NIPA-2012-H0301-12-3005). We would like to thank all of volunteers for participating in our experiment and fellows for giving high valuable comments during our research process.
The preliminary version of this paper was presented in ICONI 2012, December 2012."

http://dx.doi.org/10.3837/tiis.2012.10.006

1. Introduction

Mobile phone has become essential along with the development of wireless communication techniques. Many public institutions and private enterprises utilize the SMSs (Short Message Service) for informing or notifying their customers. However, not all of these messages are useful and people who receive unwanted spam messages often feel annoyed. According to KISA(Korea Internet Security Agency), the number of reported cases of spam is sharply increasing year on year. SMS spam is annoying for the individual and wastes there time. Furthermore, from the viewpoint of industrial productivity, it degrades the main objective of smart projects that country's governments are using in order to enhance productivity. As a mobile device is one of the core nodes needed to create a smart environment, it is vital we block junk messages on smart phones.

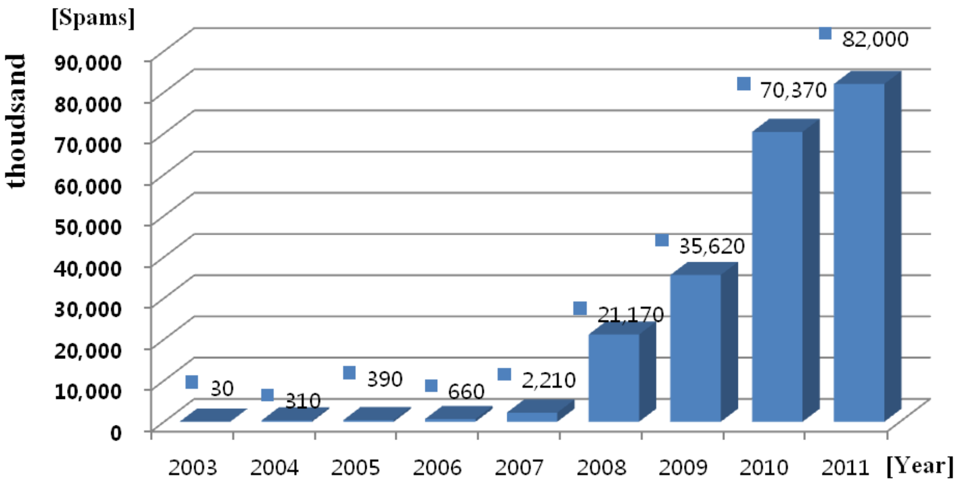

Fig. 1. Annual report of the number of received spam [from KISA]

SMS spam filters generally use two techniques: simply keyword matching or text classification. Keyword matching is easy to implement and requires little computing resources, but this needs manual word typing of blacklists or keywords. Furthermore, this method is very weak when it comes to intentional modification of keywords, such as vi@gra, b00k, B11L, so the accuracy rate for these filters is often poor. On the other hand text classification using pattern recognition algorithms dosen't require input by users. Moreover it shows better accuracy results compared to keywords matching. However, this way requires much more computation resources than filtering by keyword matching. Thus it is not easy to run text classification algorithms on mobile devices.

The purpose of this paper is to propose and examine an updatable spam filter model to reflect user preferences, which means that the filter is trained using the collected data from the user's phone and feeds back misclassified results into the training set. We expect our approach to block mobile junk messages while being user friendly and generating no additional wireless network traffic due to filtering operations.

The paper is organized in the following sections. In Section 2, we present the related work on the SMS Spam filtering problem. In Section 3, we explain the pattern recognition system from the viewpoint of text classification. Section 4 presents the system design and implementation. In section 5, we present how to collect samples and the results of experiments with our proposed system. In Section 6, we have concluding remarks and discussion.

2. Related Work

2.1 Mobile SMS Spam

Spam is advertising messages transmitted through the information network system to anonymous people without their agreement. SMS spam is annoying and a waste of time for the receiver. Moreover they hinder green growth policy of some governments, because they occupy so much network resources while transmitting to receivers. KISA, the main organization in the Korean government dealing with this kind of problem, continuously tries to cut down spam in the following ways:

- Inducing wireless network companies to develop spam filtering system on their network nodes.
- Leading mobile device product companies to load a spam management software.
- Legal regulation of spam senders by reporting their numbers.

However, in spite of the above policies, the rate of received junk messages has not dropped because of inconvenient filtering solutions that have not attracted people to use them. Moreover most filtering solutions using keyword matching are not so strong as to separate normal messages from junk ones.

2.2 The Methods To Block SMS Spam

The main nodes on the network that block SMS spam are the sender's SMS center, the receiver's SMS center and the user's device. Depending on the position of the network, each solution can adopt different methods, as shown **Fig. 2**. For example at the sender's SMS center, we can apply several methods such as 'Call and Response', 'Sending-behavior-based' and 'Text Classification'.

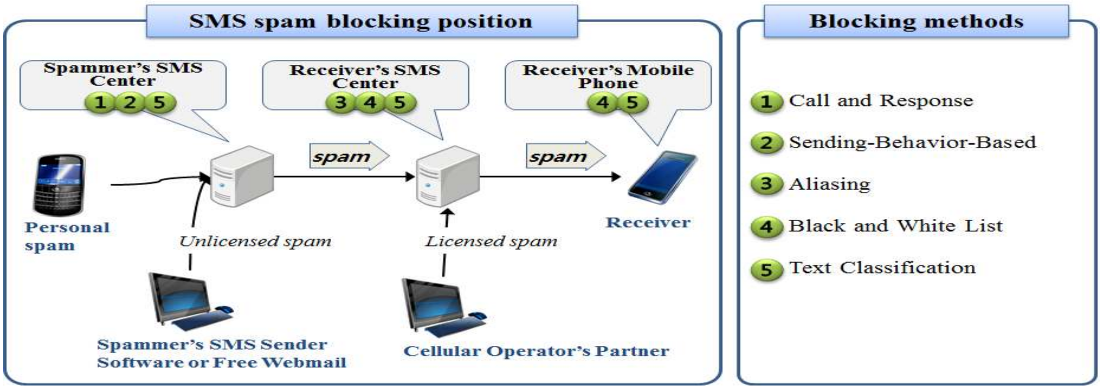

Fig. 1. Correlation between Techniques and Running Environment [1]

2.2.1 Call and Response

Call and Response is a solution at the sender's cellular operator. Every time there is a SMS sent by a user, the SMS Center will send a call and the sender must respond appropriately. The main focus of these methods is on questions that the human users can easily answer but that present computer programs cannot answer easily [2]. If the response is not appropriate then the message will be blocked, otherwise it will be passed to the receiver. For example, when a user sends a message, the SMS center may send a short piece of text in an image file, and the sender must copy the text manually in response. This call message is to make sure that the message is sent by a human, not an SMS generating machine. For the SMS spam problem, the Call and Response approach usually used is CAPTCHA (Completely Automatic Public Turing Test to Tell Computer and Human Apart) as we seen in [3][4]. The drawback of this approach is that sending a CAPTCHA image consumes cellular network bandwidth. Furthermore, some old mobile phones cannot receive an image, and hence cannot answer the question. The main advantage of this approach is that it can reduce SMS spam's effect on the consumption of network resources because SMS spam will be blocked before entering the network [3].

2.2.2 Sending-behavior-based

Spammers often display a similar pattern when sending SMS spam. They try to guess many users' phone numbers and send massive amounts of SMS to them at the same time. Hu and Yan [4] identified SMS spam by using frequent time-domain area analysis. In this technique, they collected the SMS sending system's log to analyze the time and domain features of both SMS ham and SMS spam to identify the behavior pattern of SMS spam. By inferring the pattern, they can recognize the mass sending of SMS as spam or ham. On the other hand, Qian et al. [5] classifies SMS spam based on its sending behavior using ID3. Some of the SMS's behavioral features used for analysis were sending frequency, the quantity of sending the same message continuously, and the sending success rate. This approach is maintained implemented at the spammer's SMS Center. When the SMS Center receives an SMS, it tries to detect the probability that the SMS is spam or not. The advantage of this approach is that SMS spam will not consume cellular network bandwidth, since the spam will not be forwarded to the user. However, the SMS Center blocks SMS spam based on a data set consisting of the sending behavior for SMS spam and SMS ham. Obtaining this data set is time consuming and expensive. Since the sending behavior is always changing, the blocking system must be updated continuously. Moreover, it cannot block licensed SMS spam.

2.2.3 Aliasing

As an illustration of the problem, AT&T users can receive SMS sent from the Internet. Sender, who wants to send an SMS to an AT&T user, can send message using an email to [phone number]@txt.att.net. AT&T users then receive the message on their mobile phone as an SMS. Spammers try to guess users' phone number and send them as an SMS spam via email. Spammers do not need to pay anything, since they can send SMS freely using free email services. Cellular operators give their customers alias names for his/her mobile phone number to solve this problem. Thus, if spammers send an SMS spam to the user by using free email through [phone number]@txt.att.net, the message will not be forwarded to user. The legitimate sender (non spammers) must send the message through the receiver's alias name, instead of the mobile phone number. However, the user has to tell their friends his/her alias name.

2.2.4 Black and White List

This approach is currently available as a mobile phone application. User can make a black list and white list. Black list and white list consist of the phone number of senders or keywords.

Users put senders phone numbers or keywords, such as “FREE”, “VIAGRA”, or “girls” into the filtering system. Thus, each SMS, which comes from those phone numbers or that contains keywords, will be saved into the spam folder and the mobile phone does not alert user to this incoming SMS. Usually, users use their phonebook as their white list, which means every SMS received from senders in their phonebook will be considered as SMS ham. The advantage of this approach is that users still receive all SMS and so are able to check whether there is SMS ham misclassified as SMS spam in spam folder. Black and White Lists are simple, efficient and easy to implement [6].

Moreover, cellular operators have also tried to implement black and white list based filtering systems. User can create their black and white lists and place them at the cellular operator’s server. When SMS spam comes to a user’s cellular operator, the SMS center will block the SMS based on the black and white list keywords inserted by the user. However, users can check whether there is any SMS ham blocked by the SMS Center. In Korea, KT (Korea Telecommunications) provides this service to their customer. In this case, the quality of the filtering system depends on how well the user chooses the keywords and sets up the configurations.

The disadvantage of this approach is that SMS spam still consumes cellular network bandwidth. Moreover, putting all keywords to the filtering system is time consuming. However, this approach gives mobile phone user more control over filtering SMS spam

2.2.5 Text Classification

Judging by the number of research papers, this approach gains a lot of attention compared to other approaches. This approach distinguishes SMS spam from SMS ham based on the content of the message. The Text Classification approach trains a filtering system using samples of SMS ham and SMS spam called the data set. The system can distinguish SMS spam and SMS ham after using learning the data set. It relies on the different patterns in the SMS spam and SMS ham data set. Examples of patterns are word occurrences, length, frequency of words etc. This approach benefits from using well known pattern classification algorithms such as Naïve Bayes(NB), k-Nearest Neighbor (kNN), Support Vector Machine(SVM), etc.

Joe and Shim filtered Korean SMS spam using SVM [7]. Hidalgo et al. [8] classified SMS spam from SMS ham using Naïve Bayes, C45 and SVM. They concluded that SVM is the best for their case and C45 yields worst result. Sohn et al. [9] classified a Korean SMS data set using stylistic feature such as length of SMS, average length of words, word frequencies, part-of-speech n-grams (up to trigrams), and special character such as “:)” (smiling) or “T_T” (crying). This stylistic feature allows an increase in accuracy. They see the SMS spam problem as a linguistic problem. Therefore, they do not consider the running environment.

2.3 The Weaknesses Of Earlier Study

We compare the earlier studies of SMS spam filtering in **Table 1**, we can summarize the five weaknesses in the table as follows:

- Difficult to reflect individual preferences
- Difficult to update the result of misclassifications
- Not able to avoid responsibility for misclassified results
- Easy to generate additional flows during the filtering operation
- Not free from the burden of processing in the case of text classification

Table 1. Comparison of the filtering techniques

Methods / Criteria	Call and Response	Sending-behavior-based	Aliasing	Black and White List	Text Classification
Filtering point	Sender's SMS center	Sender's SMS center	Receiver's SMS center	Receiver's SMS center, mobile device	Both SMS centers, mobile device
Filtering agent	Telco	Telco	Telco	Telco, mobile user	Mobile user
Additive traffic	Occur	Occur	Not occur	Depending on the filter point	Depending on the filter point
Individual service	Difficult	Difficult	-	Possible	Possible
Note	Need agreement of SMS senders	Need pattern log data of spam sending	-	Depending on the quality of black and white list	Need training samples and training process

- Difficult to reflect individual preferences

It is difficult to reflect users' preferences in a spam filter at the SMS center, the main node on the network. The criterion to distinguish spam from ham is different from one user to another, so only one standard could not satisfy every user's desire to separate spam from ham. Therefore, it would be worthwhile to attempt to adapt different rules for each user's spam filter in order to reflect individual user tastes.

- Difficult to update the results of misclassification

Two methods, 'Call and Response' and 'Sending-behavior-based', cannot update the result of misclassifications in the training data set. The 'Black and white list' requires the user's manual typing efforts and the error rate depends heavily on the keywords selected, so it is not easy to fit the decision margin for the filter.

- Not able to avoid responsibility for misclassified results

SMS is a fee-based service on the assumption that a sender pays money to ensure his messages will make it to receivers. If telcos try to perform spam filtering, they are not free from responsibility for misclassified messages at the SMS center. Therefore the decision-maker to utilize a spam filtering service should be individual user rather than a service provider. On the other hand, if mobile phone has independent functions to block spam, telcos would be free from the responsibility for misclassified SMSs. Moreover mobile devices don't have to connect to the network to check the spam box. It means that there is no additional traffic when identifying traffic.

- Easy to generate additional flows during the filtering operation

Filtering at the SMS center is inefficient from the perspective of network operation and management, because it releases additional data traffic into the network. 'Call and Response', 'Sending-behavior-based' can operate on the condition of exchanging information. In addition to this, traffic flows through the network containing personal information could be exposed to sniffing or hacking.

- Not free from the burden of processing in the case of text classification

Although spam filters with text classification provide usability without manual efforts, it is likely to have heavy computation and memory costs. So it is not easy to run all these processes on mobile devices. Taufiq[10] developed an independent SMS spam filter operating on a smart phone, but this approach is liable to 'the curse of dimensionality'[11] when increasing the training sample data. In addition to that his approach cannot be applied to other languages, because his model only adopts tokenizing and removing special characters for preprocessing. Therefore his approach cannot deal with other languages which require lots of dictionaries. In the case of Hangul, used for the Korean language, in order to get normalized terms after preprocessing we surely need to remove the ending word 'josa' in several dictionary entries. Joe[7] proposed a SVM spam filter model which selects meaningful features among thousands of original ones by chi-square statistics after preprocessing, but his approach does not consider the limited environment of mobile computing resources. Practically, mobile phone with limited energy, memory and computing power could not apply his approach.

3. Background

3.1 Pattern recognition

General pattern recognition systems go though several steps, as shown in **Fig. 3**. In the steps for preprocessing, the system needs to segment important data and normalize certain factors so that it focuses on meaningful information. This operation is used for making vector series that stand for the object. Feature extraction is the step to select more meaningful vectors among every vector sequence, which minimizes the complexity of the calculation and enhances the reliability of the classification. Pattern recognition algorithms are applied to the classifying step, and post-processing uses the results of the classification. So we could connect the results from the classifier to some operations that we want. After passing through several steps shown below in **Fig. 3**, we can realize intelligent systems such as speech recognition, face detection, motion recognition, text classification etc.

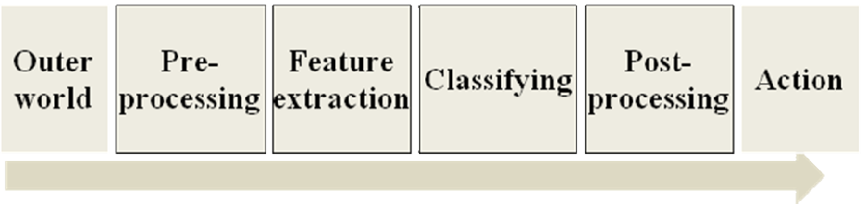

Fig. 2. General steps for pattern recognition system

3.2 The steps for text classification

3.2.1 Preprocessing

The preprocessor extracts a series of normalized factors from the original text, which means that the system can find the original form from the variously transformed words. For example if we met various types of the same word such as 'studied', 'studying', 'study' in the text, the terms are all changed to 'study', the original form. This step would be different in accordance with certain linguistic features. Usually in the preprocessing step, several methods are chosen among a lot of techniques such as tokenization, stemming, auto-spacing and normalizing numeral words, in order to acquire original form of words. In the case of Hangul, the Korean characters, preprocessing needs to build dictionaries for root words, ending words and stop words. Thus the system can remove stop words from original terms.

3.2.2 Feature Selection

Feature selection is the process of selecting a subset of all terms occurring in the training set and the system uses only this subset as key factors for classification. Feature selection serves two main purposes: First, it makes training and applying a classifier more efficient by decreasing the size of the effective vocabulary. This is of particular importance for classifiers that, unlike in NB, are expensive to train. Second, feature selection often increases classification accuracy by eliminating noise features. A noise feature is one that, when added to the document representation, increases the classification error on new data. The methods commonly used for feature selection are mutual information, chi-square statistics and frequency-based selection.

3.2.3 Classifier

The classifier categorizes incoming documents using weighted values, it is trained with a sample data set. There are several models of classifier and for binomial problems Naive Bayes and SVM (Support Vector Machine) show a good accuracy rate.

Naive Bayes as a probabilistic model is very simple and shows good performance under conditions where the occurring words are independent of each other. With this condition, the Naive Bayes classifier can classify new data only if we count the term frequency occurring in the training samples.

Support Vector Machine is a non-probabilistic classifier in which each document in the data set will be viewed as a point in |v| dimensional space. SVM draws a line in space to separate black points and white points. New incoming documents' points will be put in the space. Based on the separating line, we can classify the new incoming messages.

4. Proposed System

4.1 System description

We propose a mobile junk message filter based on contents that run on a distributed environment between a user computer and smart phone. Through this approach, we expect to overcome the weaknesses of the earlier work.

4.1.1 Configuration of proposed system

We design a distributed spam filter based on the idea of periodical synchronization between the smart phone and user computer. By synchronization, smart phones usually store important information on a user's authorized computer for backup and also update applications from the computer. Our proposed system assigns the heavy burden of the processes, such as feature selection and training, to the user computer while the smart phone takes charge of only essential jobs. Thus spam filter on the phone uses related values received from the desktop to classify new incoming messages. As shown in **Fig. 4**, the application on the phone classifies new SMS into spam or ham after simple preprocessing and vector creation. If the user regards classification by the spam filter as an error, he or she can put a tag on the misclassified message. Over the course of the synchronization process, the phone transfers the tagged errors to the authorized desktop.

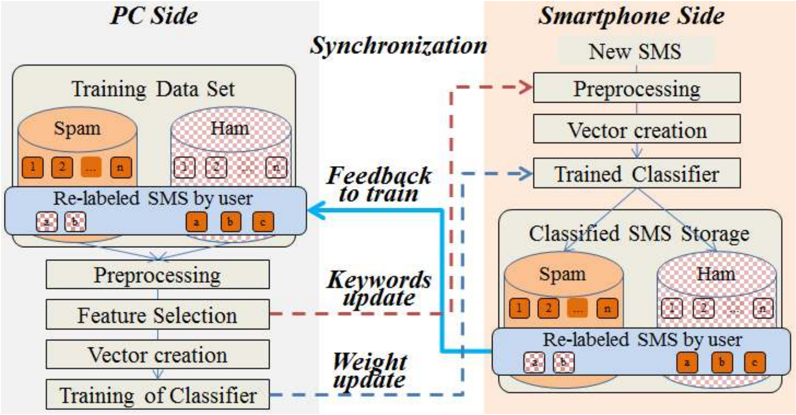

Fig. 3. System configuration

4.1.2 Features Of Proposed System

In our proposed system, the spam filter on the phone assigns several jobs that have high computational cost to the authorized personal computer, so our approach has a few advantages compared to earlier ones.

- Reflecting user preference

The classifier on the user computer modifies its weight values by the training sample data received from SMS storage on the smart phone. Until the synchronization period, the system stores the SMSs as spam, ham or re-labeled SMS. By updating the misclassified messages to the training set on the desktop, both classifiers on the phone and computer are able to renew the weight values of the decision plane.

- Consuming uniform resources

Although the number of training samples increases, the amount of work with which the smart phone has to deal with should almost be uniform. Our approach can avoid 'the curse of dimensionality'[11] by selecting a constant number of features among the thousands available from the sample set on the personal computer. The size of the vector dimension that a smart phone has to deal with is, as such, quite small and constant. So practically, a smart phone will incur little overhead when running our spam filter, even though the classifier adopts some complex algorithms like SVM.

- No additional traffic on the network

Our approach generates no additional packets while exchanging information between the phone and computer, this is because we assume that the connection between two objects is built only by the wire, not by network. Moreover our filter doesn't have to connect to a SMS center through a network to collect pattern information like in [3][4]. It runs only on the mobile phone with weight values created on the desktop.

- Free from heavy training process

The filter on the smart phone is free from burden of training process to take long time and much resource, because every job related with training progress is transferred to an authorized user computer. The smart phone only utilizes the result values which user computer created, so it can handle SVM method for classifying incoming messages without a fear of heavy training course.

4.2 Implementation

4.2.1 Exchanging information

There are three types of information exchanged between the personal computer and smart phone, as shown in Table 2.

Table 2. List of Exchanging Information

Criteria	Exchanging Information
Smart phone to PC	▪ Misclassified messages for feedback
PC to smart phone	▪ Selected terms from all terms created from the sample set ▪ Weight values for the classifier

The first type of information is the misclassified messages labeled by the user. This error data is used for retraining on the computer, so the decision boundary for the classifier can be readjusted for user preference. The second type of information, words chosen by chi-square statistics at the desktop are sent to the smart phone. This word list lightens the load in preprocessing on the phone. In the case of Hangul, Korean characters, preprocessing step is very intensive compared to other languages, because it is necessary to detach ending words, ‘josa’ in Korean, from the token. Without this detaching work, effective feature words could become dispersed, in other words the system would not be able to normalize the tokens into a standard form. Therefore the ability to distinguish between spam and ham would become greatly reduced. Moreover, as Hangul is encoded with 2 byte unicode, the average number of terms from Korean SMS is smaller than English SMS, which uses ascii code. The third type of information is the weight values for the classifier calculated with feedback data on the computer.

4.2.2 Preprocessing

Since, in the case of a Hangul document, removing ending words from a segmented term needs complex steps, we leave it to the desktop. Smart phones only need to search for keywords in new SMS using a keyword set transferred from the computer. The user computer utilizes KLT[1] (Korean Language Technology), which is a Korean morphological analyzer. A few functions of KLT are applied to preprocessing steps, such as stemming, auto word spacing, and recognizing numeric words. After preprocessing, the user’s desktop selects meaningful words that play an important role in classifying spam and ham, because all terms are from the training set, this could cause problems on a smart phone with a large memory requirement.

As shown Table 3, we removed all special characters, because they have little importance in categorizing spam and ham. Sometimes they are used to modify spam keywords on purpose. We applied auto word spacing, since most Korean spam intentionally breaks word spacing rules. Numeric words have significant characteristics, because many junk messages include prices of a loan or product. We consider numeric words to be in one of three categories: price words, only numeric words and numeric assumed words[12].

[1] KLT http:// nlp.kookmin.ac.kr/HAM/kor

Table 3. Preprocessing steps on the PC and phone

Processing Steps	PC	phone
Remove special character	✓	✓
Segment by space	✓	✓
Stemming with KLT	✓	
Auto word spacing with KLT	✓	
Keyword matching		✓
Recognizing numeric words	✓	✓
Feature selection	✓	

4.2.3 Vecto4r creation

A raw tokenized word cannot be used in the classification process. Vector creation is a process used to map a raw tokenized word into numerical data, which is then ready to be classified by the text classification algorithm. Vector creation process employs four types of data taken from the raw tokenized data:

1. The number of occurrences of words i in document j (f_{ij}).
2. The total number of words in document j (fd_j).
3. The total number of documents in which word i appears (ft_i).
4. The total number of documents (D).

There are three major techniques used for this vector creation. These techniques are *TF-IDF, Word Frequency, and Word Occurrences*. TF-IDF values represent the importance of word *i* in document *j* in all data set D. These values increase when the number of occurrences of word *i* in document *j* increases [13]. Word Frequency is the relative frequency of a word in a document. The resulting vector for each document is normalized using Euclidean unit length. The formula is depicted in (2). Word Occurrence is the number of occurrences of a word in one document as depicted by (3). We calculate this data in order to get the vector value of each word in a document.

$$v_{ij} = \frac{f_{ij}}{fd_j} \log \frac{|D|}{ft_i} \tag{1}$$

$$v_{ij} = \frac{f_{ij}}{fd_j} \tag{2}$$

$$v_{ij} = f_{ij} \tag{3}$$

Table 4. Example of Vector Creation Result

SMS ID	Type	Word Attributes				
		buy	***book***	***free***	***SMS***	***Viagra***
SMS 1	Spam	1	1	1	0	0
SMS 2	Spam	0	1	0	1	0
SMS 3	Ham	1	0	0	0	1

4.2.4 Feature Selection

During feature selection only on the desktop, the most meaningful attributes should be selected by Chi-square statistics. Without this process, a spam filter is not free from the curse of dimension[11]. In statistics, the χ^2 test is applied to test the independence of two events, where two events A and B are defined to be independent if P(AB) = P(A)P(B) or, equivalently, P(A|B) =P(A) and P(B|A) = P(B). In feature selection, two events are occurrence of the term and occurrence of the class. We then rank terms with respect to the following quantity:

$$\chi^2(D,T,C) = \sum_{Term\in\{0,1\}} \sum_{Class\in\{0,1\}} \frac{(N_{Term,Class} - E_{Term,Class})^2}{E_{Term,Class}} \tag{4}$$

N is the observed frequency in D and E is the expected frequency. For example, E_{11} is the expected frequency of term and class occurring together assuming that term and class are independent. After determining the ranking of all terms using the results of the above equation, a PC side application chooses words according to the ordered value. The chosen words play an important role in selecting the decision boundary for the classifier.

4.2.5 Training And Classification

Similar to an email spam filter, SMS spam filters using Naïve Bayes and linear SVM shows good performance[14]. Our model also adopts these two classifiers. The best class in Naïve Bayes classification is the most likely, or maximum a posteriori, class c_{map}:

$$c_{map} = \underset{c\in C}{\operatorname{argmax}} \hat{P}(c \mid d) = \underset{c\in C}{\operatorname{argmax}} \hat{P}(c) \prod_{1\le k\le n_d} \hat{P}(t_k \mid c). \tag{5}$$

In other words, Naïve Bayes regards a higher probability part as a classified result.

$$\begin{aligned} &if\ \hat{P}(c_{spam} \mid d_{SMS}) > \hat{P}(c_{ham} \mid d_{SMS}) : classified\ to\ Spam \\ &if\ \hat{P}(c_{spam} \mid d_{SMS}) \le \hat{P}(c_{ham} \mid d_{SMS}) : classified\ to\ Ham \end{aligned} \tag{6}$$

A good classification algorithm for the binomial problem, SVM is also a popular method in the area of spam filters. The principle of SVM is to secure the maximum margin between support vectors in order to figure out the decision boundary. The kind of SVM could be divided into linear or non-linear according to the kernel function. As the classification time of nonlinear SVM is proportional to the number of support vectors, we can apply linear SVM. Moreover in a recognition problem with hundreds dimension, linear SVM usually shows better accuracy than nonlinear SVM. We utilize SVM-JAVA[2] for the training and testing of our SVM. SVM-JAVA is a java implementation of John C. Platt's sequential minimal optimization (SMO) for training a support vector machine.

5. Experiments and Analysis

5.1 Samples For Experiments

We have collected a Hangul data set consisting of 3,747 pieces of SMS spam data from 41 volunteers and 3,805 pieces of SMS ham from the twitter website. Volunteers gathered their spam data automatically for six months using the spam filter system[3] provided by KT, a Korean network provider. The system operates on user keyword matching, so spam data for the experiments was collected based on various criteria. On the other hand, it was more difficult to find voluntary users to give ham data, because people don't like to make public their ham messages that may contain private information. Therefore we collected anonymous twitter messages, as these messages are very similar to ham messages written in a colloquial style. We believe that our data set is reliable for this research.

5.2 Experiments And Results

5.2.1 Accuracy And Training Time With The Number Of Attributes

In the first experiment, we performed a similar experiment to ones others have performed. We divided the data set into training data and testing data. We used 2,248 SMS ham and 2,283 SMS spam in the data set for the training process. After the preprocessing of 4,531 SMS (60%), we obtained 9,535 word attributes.

Table 5. Data Sample for Experiment

Criteria	Count	Ham	Spam
Total set	7,552(100%)	3,747	3,805
Training set	4,531(60%)	2,248	2,283
Test set	3,021(40%)	1,499	1,522

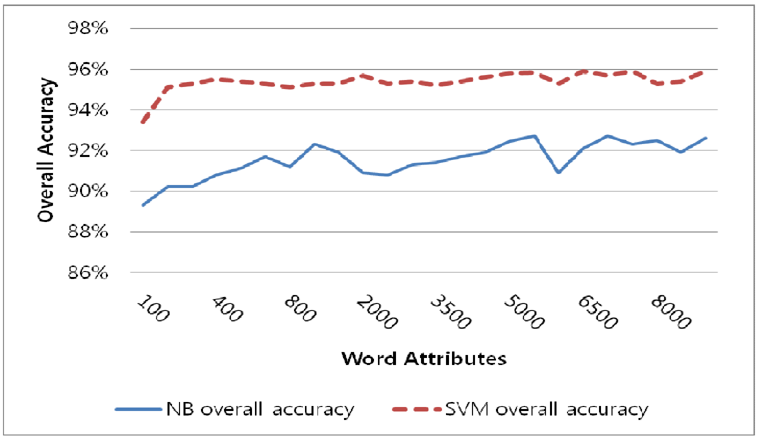

Fig. 5. Overall Accuracy with Word Attributes

[2] http://iis.hwanjoyu.org/svm-java

[3] http://mobile.olleh.com/index.asp

After the process of reducing word attributes using Chi square statistics, we observed training time and accuracy rate by Naïve Bayes and SVM.

As shown in Fig. 5, we obtained an overall average accuracy of 91.3% for Naïve Bayes and 94.8% for SVM. The accuracy was improved a little by increasing the number of attributes, but the effect for adding more attributes was mainly negative.

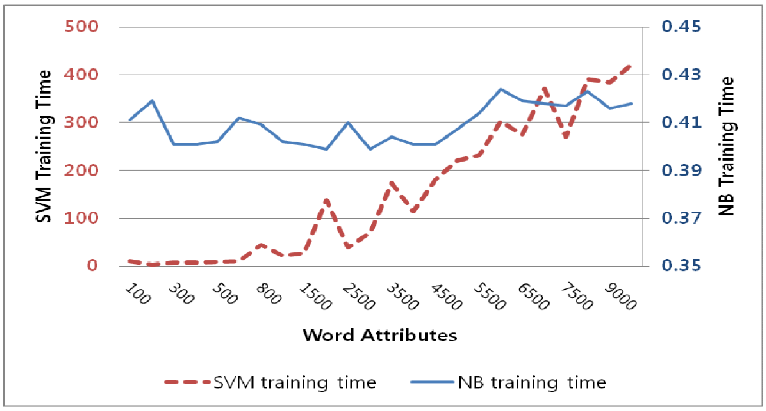

Fig. 6. Training time of SVM and Naive Bayes on user computer

In the case of training time, Naïve Bayes has more uniform and shorter times with average time of 0.41 sec than SVM, as shown in Fig. 6. Training time of SVM filter increases sharply at the point of 500 dimensions. Therefore we could say that less than 500 word attributes should be selected in order to give a reasonable response time.

5.2.2 Updating Misclassified Messages

The first experiment is not realistic, because it takes too much time to collect thousands of messages for the training set, and there is no user who would store such a large data set. Therefore, in a second experiment, we intended to simulate how our model would work in practice, as shown in Fig. 4. We assume that a mobile phone has already collected 50 spams, 50 hams for training and made periodical synchronization with the computer after classifying 50 new messages. The smart phone only added misclassified messages to the training data set on the PC and then updated information after a training process on the desktop.

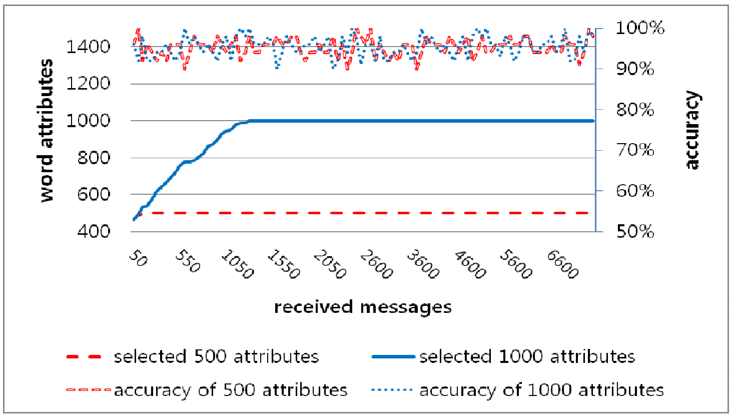

Fig. 7. Result of Naïve Bayes with feedback

As shown in Fig. 7, Naïve Bayes shows much improved accuracy result compared to the first experiment with an average accuracy of 95.1%. Using a 9 kilobyte information file, it takes about 0.08 sec to test one incoming SMS on a smart phone.

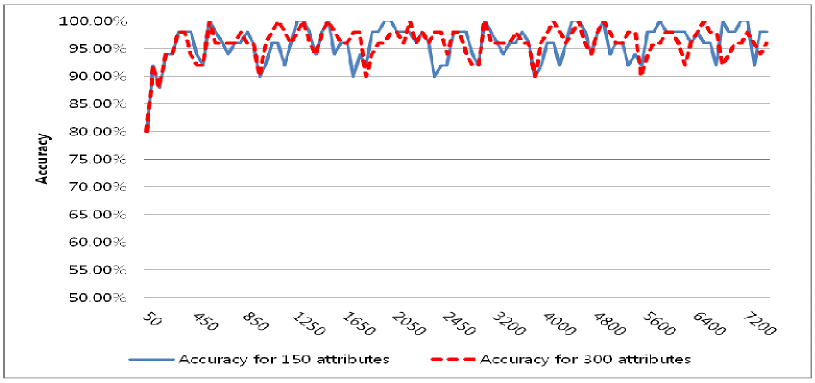

Fig. 8. Results of SVM with feedback

The SVM results, in Fig 8, show a 95.5% accuracy with only small improvements compared to the first experiment in Fig. 5. Training time is reduced to about 0.8 sec for 300 attributes and 0.5 sec for 150 attributes with only a few feedback messages. With a 3.8 kilobyte parameter file, it takes about 0.21 sec to test for one incoming SMS on a smart phone.

Both accuracy results are similar by the time 300 messages have been received, because the selected word attributes are the same. After the word attribute limit of 150 or 300 is reached, they compete to be chosen. The member words from each of the above limits is different, as such the results from the 150 attribute system is sometimes better than the 300 attribute system. Therefore, we cannot say that the 300 attribute setting is better than the 150 attribute setting in spite of better average results. We could just choose the size according to the hardware specification and usage word characteristics from the viewpoint of system design.

5.2.3 Comparison With Previous Work

In order to compare the performance of our proposed approach with previous work, we run three methods in the same environment. We used the same Korean SMS samples and run our system on a personal computer.

Table 6. Comparison with previous works

Criteria	Taufiq[10]	Joe[7]	Our model
Classifier	NB	SVM	NB, SVM
Processing	Only on a phone	Simulated model only on PC	Distributed to PC and a phone
Reflecting user preference	Feedback of misclassified message	-	Feedback of misclassified message
Limiting attributes	-	Chi-Square statistics	Chi-Square statistics
Language	English	Korean	Korean
Preprocessing	Only Tokenizing	Tokenizing, Stemming and etc	Tokenizing, Stemming and etc

We only compared the accuracy results of the three approaches when they had the same experimental conditions, because those have different properties as follows in **Table 6**. of the results of our experiments, show that our model gets higher accuracy rates than others, as shown in **Table 7**.

Table 7. Accuracy Results

Accuracy	NB	SVM
Taufiq[10]	91%	-
Joe[7]	-	92.7%
Our model	95.1%	95.5%

5.3 Experimental environment

We performed all experiments on a Google Android smart phone and a desktop computer as shown in **Table 8**. We simulated the synchronization operation by exchanging related files between the desktop computer and smart phone.

Table 8. Specifications of the Running Environment

Criteria	Specifications
Smart phone	▪ Qualcomm® QSD8250™, 1 GHz Processor ▪ Android™ 2.1 (Éclair) Operating System ▪ ROM Memory 512MB and RAM 512MB ▪ 4.MicroSD™ memory card (SD 2.0 compatible)
PC	▪ Intel core2 Duo CPU 2.93 Ghz, RAM 3.0 GB ▪ Windows 7, Java SE 1.6

6. Conclusion and Future Work

We proposed an SMS spam filter that reflects user's individual preference and requires minimal resources to run on a smart phone. Our proposed approach can apply both Naïve Bayse SVM as the classifier, the accuracy rate achieved is quite reasonable. Our model can adapt itself quickly to anyone's preference using general parameters trained with other people messages. In addition to these, our approach ensures security and privacy because synchronization happens only with the authorized user's computer.

In this paper, a smart phone traded meaningful information with only the user's computer, but it seems to be possible to substitute cloud computing for the desktop environment. However, we do incur traffic generation on the network and exposure of privacy information while transmitting. For future study, we need to verify the use of this application in the real world over a significant period of time, not simply through a simulation using collected sample data. We expect to improve the success rate of classification, if we utilize the phone book database on the smart phone to confirm legitimate senders.

References

[1] M. Taufiq, M.F.A. Abdullah, K. Kang, D. Choi, "A Survey of Preventing, Blocking and Filtering

Short Message Services (SMS) Spam," In *Proc. of International Conference on Computer and Electrical Engineering*, vol. 1, pp.462-466, 2010.

[2] M.H. Shirali-Shahreza, M. Shirali-Shahreza, "An Anti-SMS-Spam using CAPTCHA," *Proceedings of International Colloquium on Computing, Communication*, pp.318-321, 2008. Article (CrossRef Link)

[3] P. He, Y. Sun, W. Zheng, "Filtering Short Message Spam of Group Sending Using CAPTCHA," In *Proc. of Workshop on Knowledge Discovery and Data Mining*, pp.558-561, 2008. Article (CrossRef Link)

[4] X. Hu, F. Yan, "Sampling of Mass SMS Filtering Algorithm Based on Frequent Time-Domain Area," In *Proc. of Third International Conference on Knowledge Discovery and Data Mining*, pp.548-551, 2010. Article (CrossRef Link)

[5] W. Qian, H. Xue, W. Xiayou, "Studying of Classifying Junk Messages Based on The Data Mining," In *Proc. of International Conference on Management and Service Science*, pp.1-4. 2009. Article (CrossRef Link).

[6] H. Zhang, W. Wang, "Application of Bayesian Method to Spam SMS Filtering," In *Proc. of International Conference on Information Engineering and Computer Science*, pp.1-3, 2009. Article (CrossRef Link).

[7] I. Joe, H. Shim, "An SMS Spam Filtering System Using Support Vector Machine," In *Proc. of Future Generation Information Technology*, pp.577-584, 2010. Article (CrossRef Link).

[8] J.M.G. Hidalgo, G.C. Bringas, E.P. Sánz, F.C. García, "Content Based SMS Spam Filtering," In *Proc. of ACM Symposium on Document Engineering*, pp.107-114, 2006. Article (CrossRef Link).

[9] D. Sohn, J. Lee, H. Rim, "The Contribution of Stylistic Information to Content-based Mobile Spam Filtering," In *Proc. of International Joint Conference on Natural Language Processing*, pp.321-324, 2009. Article (CrossRef Link)

[10] M. Taufiq, "Independent and Personal SMS Spam Filtering," In *Proc. Of 11th IEEE International Conference on Computer and Information Technology*, Sep. 2011. Article (CrossRef Link)

[11] Christopher M. Bishop, "The Curse of Dimensionality", *Pattern Recognition and Machine Learning*, 1st ed., pp.33-38, Spinger, Singapore, 2006. Article (CrossRef Link)

[12] Seung-Shik Kang, "Classification and Normalization of Korean Numerals," *Proc. of 1999 KIISE General meeting and Autumn Conference*, pp.187-189, October, 1999.

[13] S. Robertson, "Understanding inverse document frequency: on theoretical arguments for IDF," *Journal of Documentation*, Vol. 60(5), pp.503-520, 2002. Article (CrossRef Link).

[14] Cormack, G. V., Hidalgo, J. M., and Snz, "Feature engineering for mobile (SMS) spam filtering," In *Proc. of the 30th Annual international ACM SIGIR Conference on Research and Development in information Retrieval*, ACM, pp.871-872, Jul. 2007. Article (CrossRef Link)

Kyoungju Lee is currently working for KT, the representative telecom company in South Korea. He received BS degree in Department of Computer Engineering, Chonnam National University, in 2004. He got Master degree in Department of Electronics and Computer Engineering, Chonnam National University, in 2011. His interest on research spans for context aware, pattern recognition and future internet.

Deokjai Choi is full professor of computer engineering department at Chonnam National University, South of Korea. He received BS degree in Department of Computer Science, Seoul National University, in 1982. He got Master degree in Department of Computer Science, KAIST, South Korea in 1984. He got PhD degree in Department of Computer Science and Telecommunications, University of Missouri-Kansas City, USA in 1995. His interest on research spans from context aware, pervasive computing, sensor network, future internet and IPv6. He is also currently the director of the ITRC center in Chonnam National University, a research center on mobile computing of Korea government.