An SMS Spam Filtering System Using Support Vector Machine

Inwhee Joe* and Hyetaek Shim

Division of Computer Science and Engineering, Hanyang University,
Seoul, 133-791 South Korea
iwjoe@hanyang.ac.kr

Abstract. This paper describes a powerful and adaptive spam filtering system for SMS (Short Messaging Service) that uses SVM (Support Vector Machine) and a thesaurus. The system isolates words from sample data using a pre-processing device and integrates meanings of isolated words using a thesaurus, generates features of integrated words through chi-square statistics, and studies these features. The system is realized in a Windows environment and its performance is experimentally confirmed.

Keywords: Spam filtering system, short messaging service, support vector machine, thesaurus.

1 Introduction

Mobile phones are critical communications devices, and their associated SMS is used 1.5 to 2 times as much as voice service. As SMS usage increases, spam text messages are becoming more common. The average number of spam text messages received daily was reduced from 1.7 to 0.6 from December 2004 to May 2005, but increased to 0.74 in December 2005 and then to 0.99 in March 2006. Individuals classify mobile spam text messages as annoying (32.3%) time wasting (24.8%) and violating personal privacy (21.3%). Spam filtering functions installed on mobile phones identify specific number patterns or words and recognize spam messages when those numbers or words are present. However, this method cannot properly filter every type of spam message currently being dispatched. In this paper, we describe a novel protocol for structured content based spam filtering using SVM and a thesaurus, and explore multiple ways to optimize its performance. The background, significance, and structure of this paper are described in Section 1. In Section 2, we analyze traditional approaches and propose our novel spam filtering system. In Section 3, we describe the specifications and implementation of the spam filtering system. We implement the system in Section 4. In Section 5, we analyze our experimental results and discuss their implications.

* This work was supported by the ITRC Support Program (NIPA-2010-C1090-1021-0009) and the KEIT R&D Support Program of the MKE.

T.-h. Kim et al. (Eds.): FGIT 2010, LNCS 6485, pp. 577–584, 2010.

2 Related Work

Spam filtering is a peculiar filed to automatic document classification to considering the document is spam or not. Automatic document classification means make bunch of similar documents by allocate each document to proper category by get through the classification system.. That classification is consisting of two phases. First phase is feature selection method by extracting needed feature to classify after indexing bunch of documents. Second phase is decision make process that choose right category for the result from first phase.

Automatic document classification gets ability to assign right category automatically through mechanical learning process. For this process, it tagged specific word to bunch of learned document. The word represents the documents and extracting feature means batch job to select words revealed from learned document. However if it select every word in learned document as features, it takes too much time and looses judgment. To prevent this problem, calculate weight of information for each word then select featured words for automatic classification.

In text categorization, we are dealing with a huge feature spaces. This is why; we need a feature selection mechanism. The most popular feature selection methods are document frequency thresholding (DF) [1], the X 2 statistics (CHI) [2], term strength (TS) [3], information gain (IG) [1], and mutual information (MI) [1].

2.1 Chi-square Statistics

Chi-square statistics estimate the correlation between a specific word t and a category c and determine the difference between the observed value and the predicted value. A high chi-square value increases the chance that a feature will be selected [4].

$$X^2(t,c) = \frac{N \times (AD - CB)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)} \tag{1}$$

a: number of documents containing word t among documents within category c b: number of documents containing word t among documents out of category c c: number of documents not containing word t among documents within category c d: number of documents not containing word t among documents out of category c.

This chi-square statistic has a natural value of zero if t and c are independent.

2.2 SVM(Support Vector Machine)

Standard support vector machines (SVMs) are powerful tools for data classification They classify two-category points by assigning them to one of two disjoint half spaces in either the original input space of the problem for linear classifiers, or in a higher dimensional feature space for nonlinear classifiers [5].

The role of a SVM is to construct a hyperplane as the decision surface such that the margin of separation between positive and negative examples

is maximized. This desirable property is achieved by following a principled approach in statistical learning theory. More specifically, it uses a method of structural risk minimization. The theory uses the mathematical concept of Vapnik-Chervonenkis (VC) dimensionality and states that the generalization error rate is bounded by this term.

Optimal hyperplanes are constructed so that the VC dimension is minimized. The advantage of this technique is that good generalization performance is achieved for pattern classification problems without incorporating knowledge from the problem domain. This technique can be shown to correspond to a linear method in a high-dimensional feature space nonlinearly related to the input space [6].

Moreover, even though we can think of it as a linear algorithm in a high dimensional space, in practice, it does not involve any computations in that high dimensional space. Using kernels, all necessary computations are performed directly in the input space. The kernel function maps the input vector into a high dimensional dot product feature space implicitly and is used to construct the optimal hyperplane.

3 System Design

The proposed system composes three components.

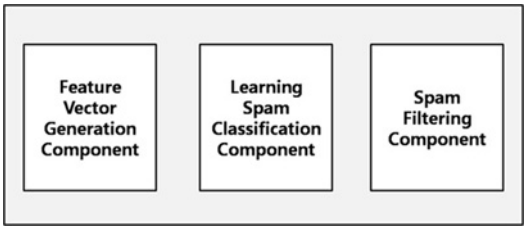

Fig. 1. Total Structure of Spam Filtering System

The first is a feature vector generator component that generates feature vectors after training. The second one is the SVM learner component using the generated feature vector. The last one is the spam filtering component to categorize spam messages using the completed classifier.

3.1 Feature Vector Generation Component

The feature vector extract component is a component that extracts feature vectors from learned data. The featured vector is a kind of array that marks 0 or 1 by the word existence. After extracting words through the preprocessor, we select the most heavily weighted word for judgment as a feature. The structure of the feature vector extraction component is as follows:

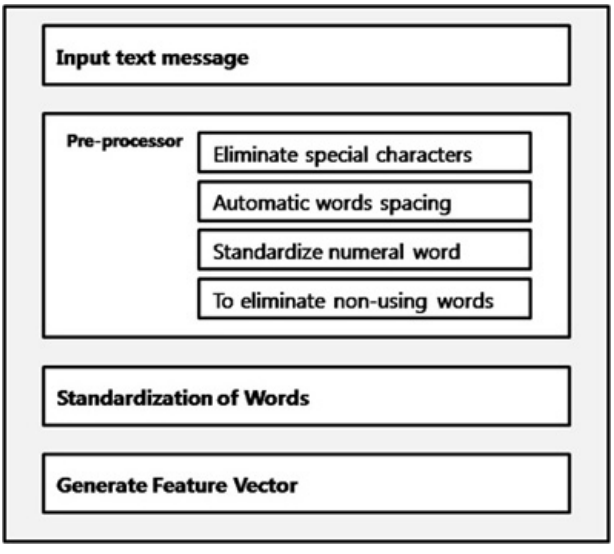

Fig. 2. Feature Vector Generation Component

The feature vector generator component goes through the following process. a. Pre-processing process b. Standardization of words using thesaurus c. Select features.

The pre-processing process then goes through the following 4 step process. - Eliminate special characters - Automatic words spacing - Standardize numeral word - Eliminate non-using words.

There are many cases in which spam SMSs include special characters. These special characters exist between words or characters, making the recognition of problem words impossible. Therefore, it is necessary to recognize words by eliminating special characters and spacing words automatically. The standardization of rhetoric and syllables recognizes both one thousand won and 1,000 won and uses it as a part of the feature vector. Non-used words are articles, prepositions, auxiliary words and conjunctions. They are eliminated since they are unnecessary. This pre-processing process is performed using the Korean language analysis module KLT [5]. The thesaurus is a word dictionary stored in a computer to search for information. It identifies special items showing synonyms, antonyms, and hyponyms. If there are synonyms among lists of words surveyed during pre-processing by the thesaurus, relevant words can be combined into one word and combined based on frequency for chi-square statistics. The feature vector was set to 100, 150, 200 and 300 in this study. The learning process was completed on each value. A higher chi-square statistic was selected as a feature and used as a learning component for the spam filtering system.

3.2 Learning Spam Classification Component

The learning component for spam classification generated learned vector data using character message data and putting it into the SVM classifier.

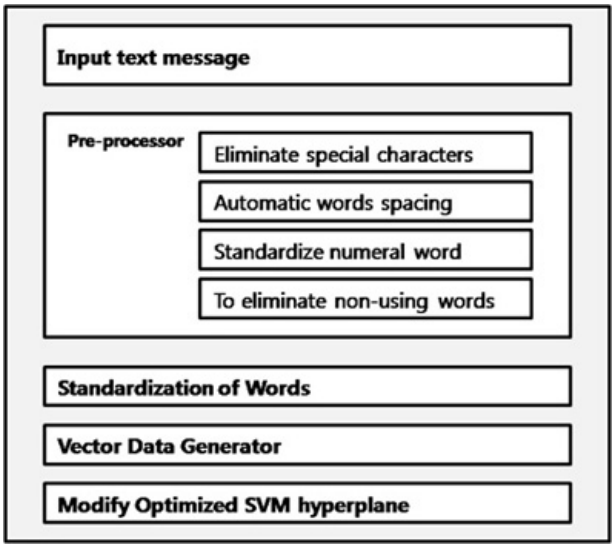

Fig. 3. Learning Spam Classification Component

The words of a text message are extracted while passing the preprocessor and standardized by the thesaurus. If the standardized word is in the feature list, the word index is set to 1 or 0. Generated vector values are used as learning data to modify the SVM hyperplane. Vector data are generated by SMS messages through two processes. If vector data has a matching word with an inserted SMS message, it marks 1 on the word. It then checks the stored contact address list to compare it with the contact address in the SMS message. If it finds a matching contact address, it marks 1 but 0. Lastly, by using information from standardizing the numeral word, if the SMS message contains money, it marks 1 or 0. After every feature vector is marked 0 or 1, a learning process is completed through SVM classifier. A Gaussian Radial Basis Function (RBF) is used as the kernel function. The constant value was set as 10, 20, 40, and gamma values were set at 0.01, 0.05, and 0.1 for this study.

3.3 Filtering Spam Component

The spam filtering component distinguishes whether the inserted data (SMS message) is spam or not by using the SVM classifier generated by the spam filter learning component. Words from the inserted SMS message are extracted by the preprocessor and standardized. Due to limitations of mobile devices, the

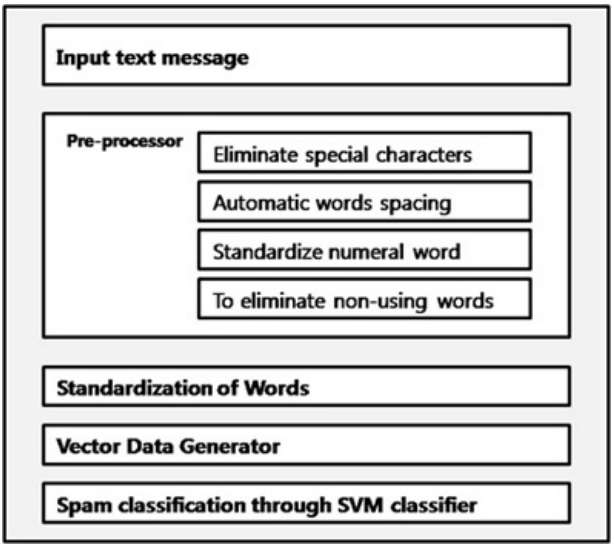

Fig. 4. Filtering Spam Component

Table 1. SPAM messages vs. Non-SPAM messages

	SPAM messages classified by system	Non-SPAM messages classified by system
Spam	a	c
Non Spam	b	d

thesaurus file extracts specific vector synonyms, and the digested thesaurus file is used for word standardization. Vector data is generated from extracted words, numeric information, and phone numbers. After putting the vector data into the SVM classifier, 0 indicates a non spam SMS message and 1 indicates spam.

4 Performance Evaluation

4.1 Performance Scaling Method

For performance scaling, this proposed system makes binary decisions in n ways following the chart below for document classification and information searching. (n = a + b + c + d)

$$\begin{aligned} \mathrm{SP} &= \frac{\text{Amount of SPAM message}}{\text{Total Amount of SPAM message}} \\ &= \frac{\mathrm{a}}{\mathrm{a+b}} (\text{if } \mathrm{a+b} > 0) \end{aligned} \tag{2}$$

$$\begin{aligned} \text{SR} &= \frac{\text{Amount of real SPAM message}}{\text{Total Amount of SPAM message}} \\ &= \frac{\text{a}}{\text{a+c}} (\text{if a+c} > 0) \end{aligned} \tag{3}$$

SP refers to Spam Precision, which is the ratio of correct to incorrect classifications among classified spam messages. SR refers to Spam Recall, which is the ratio of correct to incorrect predictions among real spam messages.

$$\begin{aligned} \text{NSP} &= \frac{\text{Amount of correct Non-SPAM message}}{\text{Total Amount of Non-SPAM message}} \\ &= \frac{\text{d}}{\text{c+d}} (\text{if c+d} > 0) \end{aligned} \tag{4}$$

$$\begin{aligned} \text{SR} &= \frac{\text{Amount of correct Non-SPAM message}}{\text{Total Amount of Non-SPAM message}} \\ &= \frac{\text{d}}{\text{b+d}} (\text{if b+d} > 0) \end{aligned} \tag{5}$$

NSP refers to Non-Spam Precision, which is the ratio of real to not real non-spam messages among those classified as non-spam messages. NSR refers to Non-Spam Recall, which is the ratio of non-spam messages classified correctly among whole real non-spam messages.

4.2 Experimental Results

Two hundred non-spam messages and 100 spam messages were used to train the system. Eighty spam messages and 80 non-spam messages, a total of 160 messages, were used for testing.

Table 2. Recognition Rate Per Number of Feature Vectors

Vector	SP	SR	NSP	NSR
100	95.89%	87.5%	88.5%	96.25%
150	93.58%	91.25%	91.46%	93.76%
200	93.24%	86.25%	87.2%	93.76%
300	86.66%	81.25%	82.35%	87.5%

Comprise recognition ratio during test with selected feature vector in higher order after apply chi-square statistic to the words. As shown in the chart, the most stable recognition result appears when the feature vector value is 150. Constant and gamma recognition tests were executed with feature vector value sets set to 150.

With a feature vector value of 150, the best level of recognition appears when the constant value is 20 and the gamma value is 0.01.

Table 3. Spam SMS Recognition Rate based on Gamma and Constant Values

Constant(Gamma)	SP	SR	NSP	NSR
10(0.01)	85.71%	75.02%	77.77%	87.5%
10(0.05)	89.33%	83.75%	84.70%	90.15%
10(0.1)	87.83%	81.25%	82.55%	88.75%
20(0.01)	91.56%	94.98%	94.80%	91.25%
20(0.05)	88.46%	86.25%	86.58%	88.75%
20(0.1)	91.78%	83.75%	85.05%	92.50%
40(0.01)	85.13%	78.75%	80.23%	86.25%
40(0.05)	87.67%	80.02%	81.60%	88.75%
40(0.1)	84.93%	77.50%	79.31%	86.25%

5 Conclusions

The spam filtering system proposed in this paper automatically sorts spam SMSs when an SMS is received based on the sender of the message and its content. The proposed system shows optimal performance with a feature vector value of 150, a constant value of 20 and a gamma value of 0.01 The proposed spam filtering system uses experience-based learning to recognize spam SMSs . Without training, it has a low recognition rate. This is a limitation of the machine learning algorithm that can be overcome by providing various patterns of learned data. The recognition rate was drastically reduced when the pre-processing device could not isolate word lines properly. Further study of automatic word spacing may be required.

References

1. Yang, Y., Pedersen, J.: A comparative study on feature selection in text categorization. In: The Fourteenth International Conference on Machine Learning (ICML 1997), pp. 412–420. Morgan Kaufmann, San Francisco (1997)
2. Schutze, H., Hull, D., Pedersen, J.: A comparison of classifiers and document representations for the routing problem. In: International ACM SIGIR conference on research and development in information retrieval (1995)
3. Yang, Y., Wilbur, J.: Using corpus statistics to remove redundant words in text categorization. Journal of the American Society of Information Science 47(5) (1996)
4. Greenwood, P.E., Nikulin, M.S.: A Guide to Chi-Square Testing. Wiley Series in Probability and Statistics. Wiley-Interscience, Hoboken (2003)
5. Cortes, C., Vapnik, V.: Support vector network. Machine Learning 20, 273–297 (1995)
6. Sahay, S.: Support Vector Machines and Document Classification (2004)
7. `http://nlp.kookmin.ac.kr/HAM/kor/index.html`