SMSAssassin: Crowdsourcing Driven Mobile-based System for SMS Spam Filtering

Kuldeep Yadav, Ponnurangam Kumaraguru, Atul Goyal, Ashish Gupta and Vinayak Naik

Indrapratha Institute of Information Technology, Delhi, India
Email: {kuldeep,pk,atul0815,ashish0814,naik}@iiitd.ac.in

Abstract

Due to increase in use of Short Message Service (SMS) over mobile phones in developing countries, there has been a burst of spam SMSes. Content-based machine learning approaches were effective in filtering email spams. Researchers have used topical and stylistic features of the SMS to classify spam and ham. SMS spam filtering can be largely influenced by the presence of regional words, abbreviations and idioms. We have tested the feasibility of applying Bayesian learning and Support Vector Machine(SVM) based machine learning techniques which were reported to be most effective in email spam filtering on a India centric dataset. In our ongoing research, as an exploratory step, we have developed a mobile-based system SMSAssassin that can filter SMS spam messages based on bayesian learning and sender blacklisting mechanism. Since the spam SMS keywords and patterns keep on changing, SMSAssassin uses crowd sourcing to keep itself updated. Using a dataset that we are collecting from users in the real-world, we evaluated our approaches and found some interesting results.

1 Introduction and Motivation

Mobile phones have become ubiquitous and pervasive in the current environment around the world. Popularity of mobile phones have increased exponentially in the last decade. One of the services that has been very popular in the mobile phones is text messaging, Short Message Service (SMS). In particular, countries like India where the usage of mobile phones have exploded (subscribers are nearly 671.69 million as of June 2010 [2]), SMS has become the way to communicate among people. SMS is the cheapest option available to reach masses. Last year, TRAI (Telecom Regulatory Authority of India) report shows that an average Indian sent approximately 29 SMSes per month [1]. Apart from personal communication, SMS is now widely used for offering value added services, advertisement medium or a mode of getting consumer involved. People have started dedicated companies for SMS based advertising solution where you can reach to 1,00,000 people in just USD 80 [3] .

Increase in SMS usage has increased the SMS spams. Now, there are various web-based solutions for bulk messaging which makes spammers' job very easy. Even after various solutions (people, process, regulatory and technology), SMS spam seems to be increasing and causing a lot of annoyance to users. SMS spam is any unsolicited message delivered to a mobile phone through text messaging. As an estimate in India, total spam SMSes per day is more than 100 million [2]. Governments and many service providers have taken various countermeasures (mostly regulatory) in order to reduce the number of SMS spam (e.g. by imposing substantial fines on spammers, blocking specific phone numbers). Like other countries, India has also set up a NDNC (National Do Not Call) registry. Mobile Subscribers have access to this NDNC registry through their operator. The main role of NDNC registry is that after registering, you are not supposed to get calls for advertisements as well as promotional SMSes [6]. It takes about 45 days for the registration to be effective. Also, the cost of a phone call is several factor high compared to an SMS [2], making SMS as a preferred choice for marketing and advertisement.

One observation is that while NDNC has largely reduced junk calls, it has failed to stop spam SMSes. There could be many reasons for the countermeasures not having an impact on the SMS spams – the operator may control the SMS spam sent by them but they don't have an automated system in place which can filter third party SMS spam; the number of mobile subscribers registered at NDNC is only 15 percent [5] of total subscribers in India. There are also other reasons like lack of penalty (e.g. monetary or suspension of the service) for Spammers and NDNC does not have a good system to report Spam problems [5]. NDNC do provide a number on which one can report a SMS spam sender identified by its mobile number but it becomes difficult when you get an SMS with numeric codes like 56789. TRAI has set a regulation in Feb 2009 that if a bulk sender wants to use small numeric codes such as a sender identification number then it has to be preceded by operator code (like "AD" to mean Airtel, Delhi).

As in email spams, there are different stakeholders who should be solving the problem of SMS spam [11]. SMS spam is nearly same as the email spam from a network operator point of view. Mobile phone network operators also have a high interest in reducing the SMS spams on the network. Because SMS spams are sent in bulk so it generates high volume of traffic which overloads the network called as SMS Flooding. SMS Flooding can delay the reception of important or valid SMS. In India, one of largest provider Airtel has stopped business of sending bulk SMSes. On the other hand, some operator use bulk SMS business to generate revenue.

In recent times, there are many media reports published on SMS spam problem [2, 3]. The most important fact here is that end-users are helpless in controlling the number of SMS spam they are receiving. SMS spam messages are annoying for the following two reasons – SMS could not be deleted without reading and it also gives a notification to user once it is received by mobile phone. Thus SMS spam wastes human attention which is most precious in information age; in some countries, you have to pay for receiving a SMS and while in roaming, you pay for all SMS that you get, so it can be very expensive even to get these SMS spams.

Unlike "email spam," the spammed user in the case of SMS does not have a provision to take any countermeasure to prevent SMS spam. Due to profileration of smart mobile devices, we can now perform spam filtering at device level thus giving a hope for

HOTMOBILE 2011 Phoenix, Arizona, USA

the SMS spams to be silently deleted without user's knowledge (*Silently eliminating the threat* kind of solution [12]). If we personalize the solution of detecting / deleting SMS spams, it may be appreciated by the users. In our user perception survey, we have found that people like to receive SMSes which contains offers. It also motivates us to develop mobile SMS spam filtering application giving full control to user what he/she wants to receive.

The contributions of this paper are:

- We present our detailed analysis on a self collected India-centric real world SMS Spam dataset which can help research community to investigate further. [1]
- Our Bayesian based spam filtering methodology detect spam and ham resonably well. We also use SVM based filtering for higher accuracy. For SVM, we have used low computation intensive feature vectors so that it can be ported to Mobile Phones easily.
- We present a proof of concept SMS inbox with Bayesian spam filtering and reporting capabilty.

The paper is organized in following sections. In Section 2, we present the related work on SMS Spam filtering problem. In Section 3, we have the listed the challenges and design goals of spam filtering system. Section 4 presents the database collection methodology and descriptive statistics about the current SMS database. In section 5, we present system description of SMSAssassin and detailed description of all three parts: Bayesian learning, mobile application and synchronization service. Section 6 presents the performance evaluation and analysis derived from our results. Section 7 presents results of our survey on user preferences about SMS spam and finally in Section 8, we discuss conclusion, limitation and future work.

2 Background

Content based filtering approaches were remarkably effective in email spam filtering. These approaches were mainly based upon machine learning algorithms which operate using some hand engineered features to differentiate a spam and a ham (legitimate SMS). The whole dataset is divided into training and testing set where machine learning approaches learns from already tagged spam and ham messages from training set. The testing set is used to analyze the effectiveness of the techniques. Researchers have tried machine learning based appproches in SMS spam filtering [7, 8, 9] . Due to short length of SMS, it is hard to find required features for machine learning classification which makes it different problem than traditional email spam [8].

Gomez et al. [7] explored the use of statistical learning based classifiers trained with lexical features, such as character and word n-grams, for SMS spam filtering. They have specifically tested the feasibility of applying bayesian based classifier to SMS spam problem. This paper also discusses the state of SMS spam in Europe and various sources of SMS spam. Cormack et al. [8] discusses the applicability of content based approaches on short messages which consist of SMS, blog comments, web logs and bulletin boards. Since, there is a problem of finding features into short message which restricts application of content based classifiers. Some of these work have focussed on expanding the feature set for content based mobile spam classifiers with additional features, such as orthogonal sparse word bi-grams [8, 9]. They were quite effective in feature vector based Machine learning algorithms like Support Vector Machines (SVM) and Orthogonal Sparse Bigrams with confidence Factor (OSBF)-Lua. OSBF-Lua exploits relationship between concatenating words by giving distance (number of words) between them; character Bigrams like ticket could be break down into "ti", "ic", "ck" etc.; character trigrams like ticket could be break down into "tic", "ick" etc.

Dae-Neung Sohn et.al. [10] took a different approach considering stylistic information and arguing that content based algorithms will not work better in some cases where common spam words like "sale", "offer" etc. may also be present in legitimate messages. So, they have selected feature set consisting of average word / message length, special character counts, function words frequencies ("the", "is") etc. to see the effect of stylistic information on SMS spam classification. The evaluation metric used by these work was Area Under Curve (AUC) in ROC curve specifially 1-AUC(%). Due to lack of common dataset/benchmark, it is very hard to compare the accuracy of two different work. However [10] shows that they have improved the accuracy produced by previous work from 10.7 to 3.8 (lower the 1- AUC(%), the better it is) using stylistic features.

Due to advent of micro-blogging, websites like Twitter also opens opportunities for spammers. In [13] authors have taken user attributes in consideration depending on its social network and content of tweets on twitter to identify spammers.

3 Challenges and Design Goals

As described in last section, [7, 8, 9] have tested the performance of email spam filters on Korean, English and Spanish dataset. They reported that direct application of these filters give low flitering accuracy due to short length of SMSes and frequent presence of short hand abbreviations, regional words etc and there is need of developing new methods of tackling this problem. Moreover, SMS spam is a comparatively less studied problem than email spam filtering; there is also a lack of common benchmark / data set for conducting studies on SMS spams. Due to presence of regional words in the SMS, we need separate dataset for each region and it will require separate feature engineering.

We present a set of design goals which will act as guidelines to design a mobile based SMS spam filtering application:

1. Computationally less intensive: The technique / algorithm used for spam filtering application should be computationally less intensive so that it can be used on mobile devices. All previous work have used a server side solution with machine learning techniques which can not be easily used on mobile devices directly.
2. Real time filtering: Detecting SMS spams in real time and making a decision on them to flag, delete, etc. will be very useful for users. By this approach, the user is not even aware of he/she getting any SMS spams.
3. Self learning: The system should keep learning from the decisions that it is making while classifying. There should also be a way to report a message as spam as well as a wrongly classified message to ham.
4. Resonable accuracy: The mobile application should give a resonable level of accuracy in filtering spam SMSes. Due to nature of machine learning algorithms, we may have some false positives and false negatives. The system should store spams so that user can check about any mis-classification.
5. Blacklisting sender: The mobile application should have provisions to blacklist a sender such that all future SMSes from that sender goes to spam folder or gets deleted (depending on the user preference).
6. Personalization: Mobile application should be personalized for the users. The prespective of users about same SMS can differ where one can see an SMS as a spam whereas someone else can see the same SMS as ham. The mobile application

[1] As far as our knowledge, this is the first research work on such a India-centric dataset and a first mobile based solution.

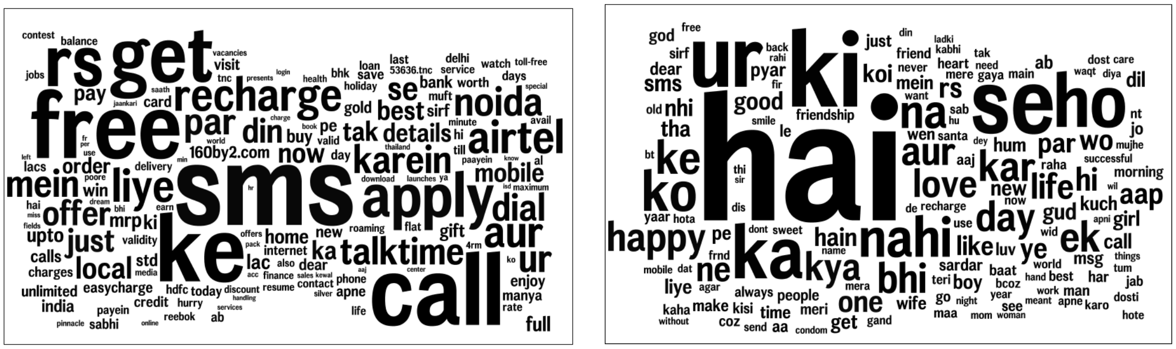


Figure 1: Tag cloud generated from the spam and ham that we collected. Left: Shows the tag cloud for spam SMSes; we see the occurrence of words like get, free, noida, apply, bhk. Right: Shows the tag cloud for ham SMSes; most of the words here are regional and not English.

should use the user preferences while filtering SMS spam.

7. Client side solution: All related work have focussed on applying machine learning techniques which are mostly implemented on a server. There is lack of an end user level mobile application for SMS spam filtering which might be good for personalization and user driven.

8. Privacy-Preserving: Since SMS is an integral part of communication now a days. Privacy is one of utmost requirement for SMS spam filtering mobile application where no SMS (private ones) of user should be revealed to third party.

9. Platform independent: There are so many hardware, software makers for Mobile devices. The mobile based spam filtering solution should be platform independent.

4 Database Collection

The SMS database that we have used for our study contains total of 4,318 SMSes. For collecting SMS spam data, we ran an incentivized crowd-sourcing scheme in our campus. There were 43 participants who forwarded spam SMSes to our SMS server to win food coupons all from New Delhi, India. Most of the participants have purchased bulk SMS credit (approx 2 USD) themselves to participate in study. We have got nearly 4000 spam SMSes in less than two months. Every fortnight, we awarded a food coupon to the participant who sent the maximum number of unique SMSes to the server. Out of 4000 SMS spam received, nearly 50% were duplicate. Therefore, we observe that most users are getting same spam SMSes. We did not collect ham SMSes through crowd-sourcing due to privacy reasons. So, we have collected ham message from close associates. We are continuing to collect data and we see an increase in the amount of data that we are collecting. We expect the SMS spam database to reach more than 20,000 in next 2-3 months. After that, we will share the database with the research community. We also plan to conduct a longitudinal study on collected Spam SMSes. Table 1 presents some of statistics about the SMS database. There were some very interesting patterns emerged if we analyze the Table 1: Average number of special character was high in hams than spam due to presence of jokes; average length of hams was also higher than spams because presence of lot of irregular spaces and long SMSes; average word length was low in hams due to frequent presence of short hand abbreviations than spams.

Figure 1 has the tag cloud generated from ham and spam messages. Due to large influence of regional words in SMSes, a SMS has a combination of Hindi and English words. We found that large portion of the SMS text in ham is from Hindi. [2] Out of the top 100 most frequent words in ham messages, 44 were Hindi language words whereas in spam words there were only 16 words from Hindi.

	Ham	Spam
Total SMSes	2195	2123
Average Length	157.6	151.6
Average number of special characters	10.2	8.7
Average number of words	29.9	23.7
Average word length	3.9	5.25
Average presence of URL	0	0.2

Table 1: Descriptive statistics of the collected data.

5 System Description

In this section, we describe the system architecture and explain each part of the architecture. Figure 2 represents the architecture of SMSAssassin. There are three major parts of the SMSAssassin: bayesian filtering algorithm, mobile application, and synchronization service that will run on a central server.

5.1 Bayesian Approach for Spam Filtering

Bayesian filtering and SVM are reported to be most succesful techniques for SMS spam filtering [8]. Bayesian filtering has been reported as one of best techniques in email spam classification systems also. Apart from that, bayesian spam filtering approach does minimal computation for classification unlike other classification algorithms like SVM, thus making it a preferred choice to use at mobile handheld devices. Like all other supervised machine learning techniques, bayesian learning also needs a seed dataset to be trained. In the training stage of bayesian learning, it computes the occurence of a word in spam as well as legitimate SMS to learn the probability of finding that word into spam / ham. For example, words like “sms”, “free”, “call” have higher probability due to their freqent occurence in spam SMSes. These words can also be seen in the SMS tag cloud shown in Figure 1. After training, Bayes theorm is used to calculate the probability of the message being a spam with different words present in that message.

After getting the probability of every word’s spaminess, the technique computes the combined probabability with basic assumption that all of these are independent events. Finally, combined probabilty value is then compared with a threshold ρ, [3] if the threshold is greater than ρ then message is likely to be a spam otherwise ham. As one can see in Figure 2, SMS Assasin uses SpamKeywordsFreq list at Mobile phones and GlobalSpamKeywordsFreq at the server to keep the track of spam keyword frequencies. In the

[2] Hindi is the official language of India.

[3] The designer of the filter can decide on the threshold value.

same way, SenderBlacklist and GlobalSenderBlacklist lists are used to detect spam based on senders' address (phone number).

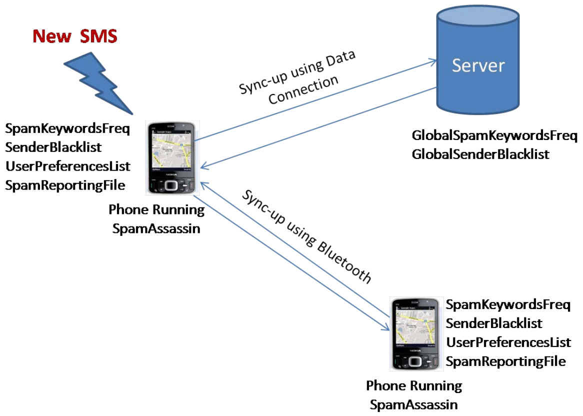


Figure 2: System Architecture of SMSAssasin.

5.2 Mobile Application

We designed and implemented a SMS Inbox with spam filtering capability as a mobile application. Mobile application implementation is done in Python S60 in Symbian Platform. The screenshots of mobile application running on a phone is given in Figure 3. One tab is for inbox messages and other tab is for spam messages. Whenever a message arrives, SMSAssasin mobile application will perform the spam filtering automatically using Bayesian filtering approach. As presented in architecture, mobile application has a list of UserPreferencesList under which two sublists are maintained for spam and ham keywords each. While computing combined probability for bayesian learning, mobile application gives higher weightage to words present in UserPreferencesList for respective decision.

Depending on Bayesian score, mobile application classifies it as spam or ham. If it is a ham SMS, then it will give a notification to the user depending on the current settings of user phone's profile, otherwise, it will silently put the SMS into spam folder. In SMSAssassin mobile application, there is also a provision to report a mis-classified SMS like email spam filters (e.g. "Not a Spam" feature in email clients). User can also report a SMS as spam if application classifies that SMS message into Inbox (e.g. "Mark as junk" feature in email clients). The following actions are taken by SMSAssasin if user reports a SMS as spam:

- Move the SMS into spam folder from Inbox.
- Give a choice to user for including the SMS sender into SenderBlackList list, it will ensure that all the future messages from this sender will go into spam folder.
- It will write the SMS content into SpamReportingFile (text file used for keeping user reported spams).
- For user preferences, it will also write the keywords of reported SMS into UserPreferencesList list.

User can occasionaly check the spam folder to see if there is a valid SMS marked as a spam (false positive error). If there is any valid SMS in spam folder then it can reported as ham. SMSAssassin will perform following steps:

1. Move the SMS to the Inbox.
2. Remove the sender from SenderBlacklist if it is there.
3. Parse the SMS content; fetch the keywords and add those keywords into UserPreferenceList.

Subsequently, user may delete the SMS from inbox or spam folder. Right now, we have implemented the spam folder capacity flexible depending on user needs where he/she can set to automatically keep on deleting spam SMSes.

5.3 Synchronization

SMS spam keywords and patterns keep on changing and spam filtering system should adapt to that change. SMSAssassin uses bayesian spam filtering approach which totally relies upon spam keyword frequencies (SpamKeywordsFreq) which needs to be updated as per current trends. SMSAssasin takes help of the server and users' social network to keep updated SpamKeywordsFreq and SenderBlacklist. In SMSAssassin mobile application settings, we have designed a parameter related to synchronization with server for getting updated version of SpamKeywordFreq. In this synchronization, SMSAssassin sends the list of reported spams to the server (SpamReportingFile). The Server will update its GlobalSpamKeywordFreq list using the ones reported by the user. The server has to just update the GlobalSpamKeywordFreq list based on keywords in reported spam, so it can be done in real time. If many users like a crowdsourcing system contribute their SMS spam to GlobalSpamKeywordFreq list, the spam classification accuracy of the SMSAssassin gets increased.

The synchronization can also happen between two independent mobile phones (independent users) using Bluetooth. For instance: user A and user B have SMSAssassin mobile application running on their mobile phone. They both can share their reported spam list to each other and update their SpamKeywordFreq list for better filtering. In the same way, SenderBlacklist can also be shared to block spammers based on the sender address.

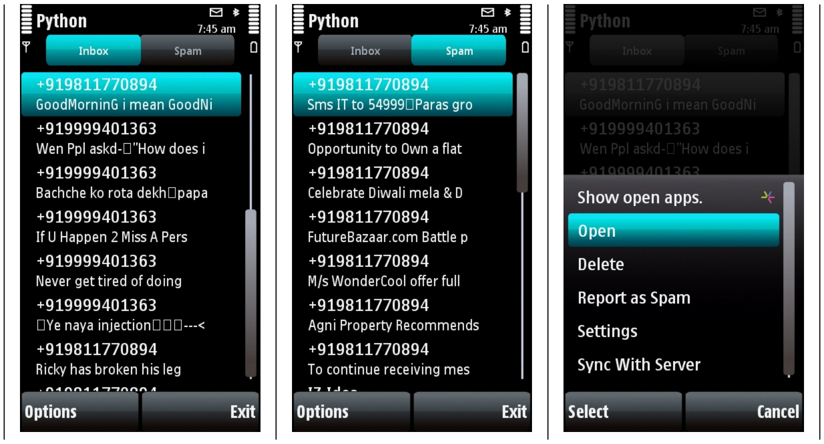
We hypothesize that above described crowdsourcing based synchronization techniques could be extremely effective in improving accuracy of SMSAssain. For instance, in festive seasons there will be multiple SMS spams coming with distinct keywords like "Diwali offer"' which may not be present in system before hand. [4] Whenever one user reports it as spam and sync SpamKeywordsFreq list with the server, this can be synchronized with all other users and thereby benefit other users. We want to emphasize on the fact that the user's privacy is not at stake during any of above synchronization because the application does not send any personal information to the server. It just sends a list of reported Spam SMSes (SMSReportingFile) which itself is of no use to him/her.

6 Performance Evaluation and Analysis

We have divided the whole dataset into training and testing set. The training set contained 1000 ham and 1000 spam messages. We have tested the bayesian learning accuracy with rest of the dataset which contained around 1195 ham and 1123 spam messages. In testing dataset, bayesian learning gives 97% classification accuracy in ham and around 72.5% classification accuracy in spam SMSes. All bayesian learning results are produced using a running SMSAssasin mobile application in Nokia 5800. Although, a pre-generated training file i.e SpamKeywordsFreq (using a Desktop application) was provided to the mobile application. At an average, Nokia 5800 took 0.61 sec in deciding a SMS as spam or ham using Bayesian learning. Table 2 presents the classification accuracy of different machine learning techniques on our dataset.

Bayesian produced comparatively low spam classification accuracy because it failed to recognize the different abbreviations of word in spam SMSes like Xtension (referring Extension), chrgs (referring charges) etc. Also, there are some words which had frequent occurence in both spam/ham like reply,bar etc. By good ham

[4] Diwali is one of the popular festivals across different parts of India.



+919811770894
GoodMorninG i mean GoodNi
+919999401363
Wen Ppl askd-□"How does i
+919999401363
Bachche ko rota dekh□papa
+919999401363
If U Happen 2 Miss A Pers
+919999401363
Never get tired of doing
+919999401363
□Ye naya injection□□□---<
+919811770894
Ricky has broken his leg

+919811770894
Sms IT to 54999□Paras gro
+919811770894
Opportunity to Own a flat
+919811770894
Celebrate Diwali mela & D
+919811770894
FutureBazaar.com Battle p
+919811770894
M/s WonderCool offer full
+919811770894
Agni Property Recommends
+919811770894
To continue receiving mes

Figure 3: Snapshots of running SMSAssassin Mobile Application in Nokia 5800 phone in PyS60 environment. Application has two different tabs : Inbox and spam. User is able to report any misclassified SMS as spam or ham.

	Ham Accuracy	**Spam Accuracy**
Bayesian Learning	97%	72.5%
SVM	93%	86%

Table 2: Classification accuracy comparison of machine learning approaches for SMS spam filtering using same training and testing set.

classification produced by bayesian, we can say that there is very low probability that legitimate(ham) messages will be classified as spam.

Motivated by low spam classification accuracy by Bayesian, we have used Support Vector Machine(SVM) for classification. SVMs are supervised machine learning algorithm where using training data it builds a classification model. That classification model is used to predict the category of new examples or testing data. We have used SVM implementation of SVMLib library for classification in our dataset. For SVM to give good accuracy, we need to find good features set which can be a good discriminator in spam and ham. We have started with a set of 20 features which contains keyword based features such as presence of spam words and phrases and stylistic features like presence of special characters, average word length etc. We have taken special care in selecting features which were less computation intensive so that it can easily be performed at a mobile appication. Using SVMLib, we have computed F-score for comparing performance of various features in classification. F-score shows importance of features i.e higher value of F-score means that feature is a good discriminator of spam and ham. In Table 2, we have presented the top 10 features with their respective F-score in our dataset.

There are following interesting observations that came out from our features engineering on a India centric dataset.

1. There are some special characters like '/' which are very frequent in spams.
2. Average word length in most of spam SMSes is high due to presence of special character in place of space. Also, average word length is low in most of hams due to use of frequent short hand abbreviations.
3. We have found that the SMSes which were having a URL were mostly spam.
4. Spam SMSes have a higher probability for numeric words because each spam SMS contains a number to call or SMS.

Feature	**F-score**
Count of Spam words	0.141353
Count of '/'	0.138823
Average word length (space as delimiter)	0.131290
Average word length (space and special characters as delimiter)	0.118367
Presence of URL	0.116232
Count of numeric words	0.098747
Count of alpha-numeric words	0.098747
Presence of Smileys	0.096299
Presence of full URL	0.096299
Number of words	0.063765

Table 3: Top 10 features used for SVM classification with their F-score.

Based on the 20 different features, SVM improved upon the spam classification accuracy produced by Bayesian learning (refer Table 2). Surprisingly, SVM decreases the ham classification accuracy. We have noticed that it happened due to lot of machine generated ham SMSes and also the discriminating features such as special characters were also present in some of ham SMSes(mostly jokes). Even while using low computational intensive feature set in SVM, we were able to get comparable accuracy where our 1-AUC(%) value was 5.61.

7 User Preferences

Decision for SMS being a spam/ham may differ from person to person. For finding global patterns in user preferenes with respect to gender, age and type of spam SMS, we developed a "SMS spam user study"' web application. We have randomly selected 200 spam SMSes from our SMS spam dataset for this study. Most of these 200 SMS (Intially collected by crowdsourcing as Spam SMSes) were marketing messages and main aim for this study was to get user perceptions about those SMSes. Each participant was presented 10 different spam SMSes with four different options to choose from : spam, ham, not-sure, skip. There were 39 male and 11 female participants in the study. There were total 477 replies for total 200 SMSes after the study. Average age of female participants was 23.5 years whereas for the male, it was 21 years. There were two interesting observations which came out from this study:

Females mostly tagged message related to festivals offers, beauty products, discount offers on food/goods etc as ham; males mostly tagged messages related to sports, businesses(opening of new shop etc), job, studies, social gathering and recharge schemes related messages as ham. Further, We have observed that most of food discount offers were tagged as ham by males and females both.

8 Discussion

SMS spam filtering is an important problem to solve and make use of the of the Information Communication Technologies (ICT) to the fullest. We have designed a bayesian based Mobile Spam filtering application which satisfies most of design goals with resonable accuracy. Considering user perception about spam SMSes, we have provided a user oriented solution where different tabs in mobile application gives user freedom to receive SMSes which are spams but still useful to him/her. Reception of SMS does not cost in India even in the roaming, this kind of solution may work well.

Since data plan on the phones is still not used by masses in India, we have kept synchronization time user specific and provided him/her option to modify it. Even if user does not use synchonization service, bayesian learning based SMSAssassin can be used as a stand alone application on the phone.In a developing country like India, there are lot of programmable mid price range phones (100$ to 400$) on which SMSAssassin can be deployed easily. We plan to conduct a user study of SMSAssasin by deploying it on those phones. Since the basic idea of SMSAssasin does not make it a platform dependent solution, we are developing mobile application for Android and Windows Mobile based phones also for wider applicability of the system.

In Section 6, we have shown that bayesian learning gives low spam classification accuracy and SVM has improved upon the spam classification accuracy with a resonable ham classification accuracy. Due to its high computation requirement, SVM classification cannot be performed at mobile application level. As an ongoing work, we are developing a server based mobile system to perform SVM classification online. In this mobile system, whenever a SMS arrives on to the phone, our application captures that SMS without showing it to the user and fetch features values using SMS content. For SVM classification, We have used all the light weight features like word length, presence of special characters etc which can easily be computed by a mobile application. Mobile application then sends those feature values to the server which performs classification. The server will then run a pre-trained classifier which can be occasionally updated using user contributed spam messages. Server will also take a very little time in classification of the SMS and will send the result back to the mobile device. Also, here server could be replaced by a cloud where mobile application will interact with cloud to make a decision on SMS. One thing to note here is that user's privacy is not at stake; because mobile device just sends feature vales (just an integer vector) instead of SMS content. However, we still have to think a way to get user preferences in account while performing SVM based classification.

As we have mentioned before, this is an exploratory and ongoing work, we present following future research directions:

1. We plan to test effectiveness and accuracy of SMSAssassin in real-world by deploying it among mobile users; we also plan to evaluate empirically the effectiveness of crowd sourcing – how much it contributes to the accuracy of a spam classifier?

2. From our survey of user pereceptions about SMS spam, we have found that perception of more than one user about the same SMS may differ. We would like to extend SMSAssassin to a user specific personalized mobile based spam filtering system.

3. Considering high accuracy of email spam filters, extensive feature engineering is required to select good features which are readily computable at mobile devices as well as improve accuracy of classification. Also use of online machine learning classifier can be explored for increasing the accuracy for SMS spam filters.

4. SMSAssassin uses topical and stylistic features to classify spam/ham which can be mis-guided by spammers, there is need to study thread models to this scheme.

9 Acknowledgements

We would like to thank all the participants who participated in the study and data collection. We are thankful to anonymous reviewers who have given constructive feedback to improve the paper. The authors would like to thank International Development Research Centre (IRDC) and all members of PreCog and MUC research groups at IIIT-Delhi. We would like to acknowledge the inputs of Dr. Ashish Sureka and Dr.Richa Singh in the early versions of this paper.

10 References

[1] http://trak.in/tags/business/2009/07/07/full-report-sms-vas-usage-india/

[2] Times Of India. http://timesofindia.indiatimes.com/tech/personal-tech/computing/Junk-SMS-No-end-to-mobile-spam-mess/articleshow/6247207.cms

[3] http://www.livemint.com/2010/07/27000020/Scourge-of-SMS-spam-swamps-mob.html

[4] http://www.medianama.com/2010/09/223-sms-spam-increase-india/

[5] http://emergic.org/2010/08/09/ending-sms-spam-part-1/

[6] http://ndncregistry.gov.in/ndncregistry/index.jsp

[7] Hidalgo et al. Content based SMS spam filtering. In Proceedings of the 2006 ACM Symposium on Document Engineering (Amsterdam, The Netherlands, October 10 - 13, 2006).

[8] Cormack et al. Feature engineering for mobile (SMS) spam filtering. In Proceedings of the 30th Annual international ACM SIGIR Conference on Research and Development in information Retrieval (Amsterdam, The Netherlands, July 23 - 27, 2007).

[9] Cormack et al, Spam filtering for short messages, Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, November 06-10, 2007, Lisbon, Portugal .

[10] Sohn et al. The contribution of stylistic information to content-based mobile spam filtering. In Proceedings of the ACL-IJCNLP 2009 Conference Short Papers.

[11] Sheng et al. Improving phishing countermeasures: An analysis of expert interviews. APWG eCrime Researchers Summit (2009).

[12] Kumaraguru et al. Teaching johnny not to fall for phish. ACM Transactions on Internet Technology (TOIT) 10, 2 (2010).

[13] Benevenuto et al, Detecting Spammers on Twitter, CEAS 2010 Seventh annual Collaboration, Electronic messaging, AntiAbuse and Spam Conference July 1314, 2010, Redmond, Washington, US.

[14] Kuldeep et al. 2010. Challenges and novelties while using mobile phones as ICT devices for Indian masses: short paper. In Proceedings of the 4th ACM Workshop on Networked Systems for Developing Regions (NSDR'10)