Contributions to the study of bi-lingual Roman Urdu SMS Spam filtering

Kashif Mehmood[1], Hammad Afzal[2], Awais Majeed[3], Hassan Latif[4]

[1,2,4]National University of Sciences and Technology Islamabad, Pakistan
[3]Bahria University Islamabad, Pakistan
[1]kashifmehmood.mscs20@students.mcs.edu.pk, [2]hammad.afzal@mcs.edu.pk, [3]awais.majeed@bui.edu.pk,
[4]hassan.latif@live.com

***Abstract*—With the increased usage of internet and mobile phones, number of spams has also increased in both these areas. The Spam in both these areas is an increasing threat and sometimes cause huge financial as well as data/confidentiality loss. Therefore, actions need to be taken to stop these spams on both media. This paper analyses various techniques that are currently being used in Spam filtering in the context of mobile text messages. The contents of SMS are unique in nature so some techniques might be effective while some might not be. Some of mostly used algorithms and techniques are discussed in this paper. Furthermore, we have performed automatic spam filtering using machine learning algorithms on Roman Urdu text messages and achieved an accuracy of 92.2% on a manually curated corpus of 8449 messages. The SMS corpus has also been made available for future research works.**

Keywords:SMS, ham, spam, arff, WEKA, JAVA,roman urdu

I. INTRODUCTION

With the increased usage of Internet and emails, mobile phones and Short Messages (SMS) usage is also increasing which gave rise to spam in both these areas. The surge of spams in mobile phones through SMS is a growing problem and actions need to be taken to stop these unwanted messages. Our research analyses a few techniques that are currently being used in spam filtering, particularly in the context of SMS and generally for longer messages (emails etc). The contents of SMS are unique in nature so some techniques which are efficient for longer messages may not be as effective while for shorter messages. Furthermore, messages on tweets are also limited to only 140 characters, therefore, the characteristics exhibited in mobile SMS can be related to those in tweets as well.

Spam is defined as any message or content sent to the user without his/her consent that he/she does not want or needs. US Federal Trade Commission (FTC) defines spam as any commercial message sent without the consumer's consent or request [1] .Short messages (SMS) and emails were used as a medium for spam due to the large number of cellular subscribers and internet users. During last decade, SMS has emerged as a popular medium of quick communication. The revenues from SMS are also growing. In the US, the revenues from 2010 were US$ 105.5 billion which are being forecasted to rise to US$136.9 billion till 2015.The global SMS traffic is 5 trillion messages and is expected to rise to 8.7 trillion SMS till end of 2015. In the US alone the SMS traffic was 2.3 trillion and is expected to rise to 3.5 trillion in 2015 [2]. A cost of an SMS can be as low as US$ 0.001 and even free in some countries like China. The low cost communication medium has automatically become target of spamming. According to Cloudmark stats, spam on mobile phone varies from region to region. For example, in North America less than 1% of messages are spam while in the parts of Asia; the percentage is up to 30%.

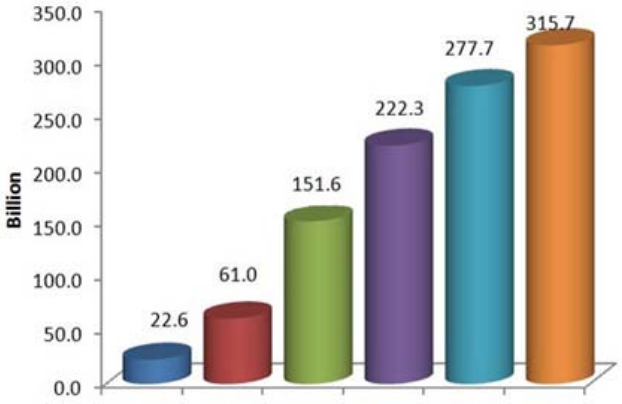

Fig.1. SMS Statistics from Pakistan (Image Courtesy PTA [3])

Similar is the trend in developing countries like Pakistan which has vast reach of mobile operators throughout the country. The cost of SMS is much lower where one can get a bundle of 5000 messages just for 5 Rupees (0.05 USD). Due to this low cast medium, a large number of people in advertising and marketing are using it for their benefits which can be annoying for users. Some people also use it for spamming purposes. In Pakistan 301.7 billion text

messages were exchanged from July 2013 to June 2014 which is almost 800 million text messages a day [3]. With such vast exchange rate of SMS, users are prone to the spammers who also have adopted SMS for spamming purposes.

Although mobile and computers are popular in younger generation, most of these resources are equipped with UK/US keyboards. This has resulted in popularity of Roman Urdu (Urdu written using Latin alphabets) which is typically used as language of communication on mobile phones (SMS) as well as on social media (Twitter, Facebook). A number of studies have been carried out for Roman Urdu such as sentiment analysis [4][5][6] on tweets and Spam detection in tweets [7].

In this research we have collected a corpus of Roman Urdu SMS containing 50,000 messages, labeled by Amazon's M-Turks for class labels Spam and ham. The dataset is available at [9]. However, corpus contained a lot of noisy data with duplicate values and some meaningless incomplete sentences. We processed the corpus with the help of domain expert and manually tag it for Spam/Ham. Using different text based features including Bag-of-Words, Term frequency and n-grams along with the classifiers such as Support Vector Machine (SVM), Decision Tree (DT), we managed to achieve an accuracy of 92.2% in spam detection.

The rest of the paper is as follows: Section 2 contains related work that shows various spam filtering techniques used in bi-lingual text, Section 3 presents methodology followed by Results and Discussion in Section 4. Finally, paper is concluded in Section 5 with directions for future work.

II. Related work

This paper provides a detailed view of relate work.

Longzhen and Longjun [9] has discussed some of spam filtering approach using the combination of both the K-Nearest Neighbour classification and Rough Sets to separate spam from legitimate messages. Rough set is used to remove redundant and un-necessary attribute from the data to improve classification accuracy. Reduction of features makes the decision making real quick. The dataset they are using have 550 spam messages while 200 normal text messages, a total of 750 message. The authors used precision, recall and F-measure as performance standards. By using k=12, they got precision of 91.35% while a recall of 84.50%.

Content based approaches are assumption based and they read the whole text of a message in order to classify the message as Spam or ham [10]. It analyses the SMS using features (tokens) and then decide it whether this SMS is legitimate or Spam. Content based filtering are of two kinds. Statistical based and rule based.

The Bayesian Learning Approach was proposed by Zang and Wang [11]. It makes use of the Bayesian learning algorithm and its use on SMS filtering in mobile text paradigm. Word segmentation and tokenization is carried out through ICTCLAS (Institute of Computing Technology Lexical Analysis System) and then Bayesian approach is applied on those segments for classification. A training set is provided and on the basis of that training set, an unseen message is classified as spam or ham. Bayesian rule is mostly used in classification due to its quickness and low resource consumption in classification.

$$p(C_k|x) = \frac{p(C_k) \times p(x|C_k)}{p(x)}$$

Bayesian Rule

In the equation, p(x) shows the a-priori or beforehand probability of a message having the vector x, and $p(C_k)$ shows that a message belong to the class C_k. If we calculate $p(C_k)$ and $p(x|C_k)$, then we can deduce $p(C_k|x)$ Bayesian rule will then classify according to following rule. P(Spam|x)>p(Ham|x).The probability that the specific message containing the vector x belongs to the spam class is greater than the same message belonging to the Ham (non-spam) class.

In [12] Mathew and Issac collected a corpus of 5000 SMS out of which 15% are spam messages. The authors used WEKA [13] for experiments. As most of the algorithms cannot process text so they used 'StringtoWordVector' feature of WEKA which converts the text into word vectors. Best accuracy of 98.2% was achieved by Naïve Bayesian Multinomial, a variant of Bayesian algorithm. The authors preferred "Discriminative Multinomial Naïve Bayesian (DMNB) Text" over Naïve Bayesian because this algorithm returned an accuracy of 97.2% while giving the lowest false positive rate.

Hidalgo et al. performed Bayesian filtering on SMS by collecting different SMS collections from different sources [14]. They collected two different corpuses for two different languages, Spanish and English. For English the authors combined SMS from John Stevenson Corpus (JSC), National University of Singapore NUS SMS Corpus and a UK based forum Grumble text and made a corpus of 1119 legitimate messages and 82 spam messages. The authors performed feature selection on this corpus and selected attributes which scored more than 10 by using Information Gain (IG) [15][16]. Their dataset contained a total of 1203 messages where 82 were spam while 1119 were ham messages. The machine learning algorithms used were

Naïve Bayes [11], C4.5[17], PART [18] and SVM [19]. The authors found out that SVM is a best machine learning algorithm in their scenario as it produced less number of false positives whereas performance of C4.5 was the lowest.

Almeida and Hidalgo [20] reported another work in same domain where they collected SMS from individual and other sources and made a corpus of SMS that is also publically available. 425 messages were taken from Grumble Text website. 3375 messages were randomly chosen from NUS SMS corpus. 450 SMS were taken from Caroline Tag's PhD Thesis available online. 1324 messages were taken from corpus available online. The corpus made from all these sources is publicly available for researchers. The new collection is composed of 4827 legitimate messages and 747 spam messages, a total of 5574 messages. The authors claim that this is the largest corpus that existed at the time of their work. The authors divided the corpus in two parts. One for testing and another for training. 3900 messages were used for testing and 1674 were used for training. They used many algorithms but they found out that SVM and Naïve Bayesian performed best in their case with an overall accuracy of 97.64% and 97.50% respectively. Apart from accuracy the authors also used Mathew's Correlation Coefficient [21] for classifier performance.

III. METHODOLOGY

Our methodology comprises of following steps

1. Data Pre-processing
    a. Removal of special characters.
    b. Removal of very short meaningless messages.
    c. Removal of similar messages.
    d. Conversion to small letters.
2. Data Preparing for Classification.
3. Spam Filtering (Data Classification)

A. Data Pre-processing

SMS are fed into a program for pre-processing steps such as removing non alphanumeric characters which do not mean anything in spam context. SMS of size less than 6 are also removed as they do not give much information to the classifier. While removing duplicate messages, not only the exact matching was considered by also messages similar up to a certain degree were also considered as same. For this purpose, a threshold value of 50% was used, i.e. all messages that were 50% or more alike were removed because our initial results indicated that these kind of messages were resulting in a lot of false positives. Finally, all messages were converted into small letters. Table 1 shows few examples of messages in our corpus.

Table 1: Snapshot of data

TEXT	IS_SPAM
plz write follow saman1 without with 172 folowr send zong 2323 ufone 414 warid 9536 thnx loverr	YES
mujhe kia saza sunai thee hahahaha	NO
Keh rahe she very nice pehle kiun nahen btaya	NO
inews3 updetpak overs note news Pakistan tamam shehr on sir fine wssms send krtai news 00000000000inam 00000000000	YES

B. Data preparation for Classification

After preprocessing of data, SMS are then labeled as spam or ham (legitimate) by the domain experts. Final dataset consisted of 8449 total messages with 7267 ham and 1182 spam messages. After labeling, messages are converted into format that is accepted by WEKA.

The algorithms that we were interested in do not apply on text instances so we used 'StringToWordVector' function of WEKA which changes text to numeric instances. After this conversion, almost all algorithms except those which run on binary attributes can be applied to the dataset. This function generates word vectors for classification. One example of StringToWordVector' function is shown in table 2. This method assigns a numeric identity to each word and also stores the number of occurrences (frequency) of the word in that message. We also used Bagging as feature generation step.

Table 2: Vectors Representation

Text	Features (Words)	Vector Representation
mujhe kia saza sunai thee hahahaha	Mujhay Kia Saza Sunai Thee hahaha	{3 1,4 1,6 1,9 1,12 1,14 1}

C. Spam Filtering (Data Classification)

Weka is used in this research to perform classification of tweets.

We used following algorithms in our study:

- Naïve Bayes Multinomial
- DMNBText
- LibSVM
- Liblinear
- Sequential Minimal Optimization (SMO)

Naïve Bayes Multinomial is a variant of Naïve Bayes with multinomial distribution. Multinomial is the generalization of binomial distribution. The multinomial distribution gives

the probability of any combinations of number of successes for various categories. While binomial is the probability distributions is the number of successes for one of just two categories. Naïve Bayes multinomial calculates the probability of number of successes for all the categories but in this search there are only two categories so it will be binomial distribution.

DMNB is Discriminative Multinomial Naïve Bayes. It is also a variant of NB and works on the same Bayes rule. It was suggested and experimented by Su Jiang and others [22]

SVM works on feature vectors. Suppose there are two classes in which classification is to be performed, SVM creates a vector and decision boundary in an x-y plane which contains vectors and class labels on the basis of which a certain document is classified into each one of the classes.

Liblinear is a variant of SVM with different kernel implementation. It a library for linear classification of large documents. As opposed to its origin SVM, Liblinear is quite fast and works well on large documents as shown in our results as well. In this research, regularized loss support vector machine type has been used for Liblinear [23].

SMO is Sequential Minimal Optimization which is a variant of SVM with different kernel implementation. In this research poly kernel implementation have been used with SMO. SMO is also fast as compared with its original SVM.

IV. RESULTS AND DISCUSSION

The results of classification are presented in this section. Results have been produced using 10-fold cross validation. For measuring algorithm's performance different measures have been used. They are accuracy, ROC AUC. Accuracy is the number of correctly classified instances as compared to the incorrectly specified. ROC is Receiver Operator Characteristics and AUC is Area Under Curve. ROC curves are used to measure the performance. More is AUC; the better is classification performance of the algorithm.

SMO showed highest accuracy of 93.3% while taking a time of 81.6 seconds. DMNBText showed an accuracy of 92.74% while taking time of just 0.13 seconds while Naïve Bayes Multinomial showed an accuracy of 92.22% while taking time of 0.08 seconds. Being a variant of SVM,Liblinear showed an accuracy of 91.42% while taking time of 0.58 seconds. All the algorithms returned an accuracy of more than 90% except SVM which returned 88.42%.

Naïve Bayes Multinomial showed an AUC of 0.913. SVM showed an AUC of 0.5 which is very poor. SVM formed linear curve which means that the prediction is not done systematically but abnormally or randomly. SMO showed an AUC of 0.807 which is still better than its origin SVM. Liblinear showed an AUC of 0.759. Results are summarized in Table 3.

TABLE3: Performance comparison of algorithms

	Time taken (seconds)	ROC AUC	Accuracy (%)
NB	0.08	0.913	92.22
DMNBText	0.13	0.912	92.74
LibSVM	13.3	0.5	88.42
SMO	81.6	0.807	93.33
Liblinear	0.58	0.759	91.42

From Fig 2 ROC curves of different algorithms can be seen.AUC of NB Multinomial is 0.913 while that of DMNBis 0.912. They both are similar but distract going towards 1.

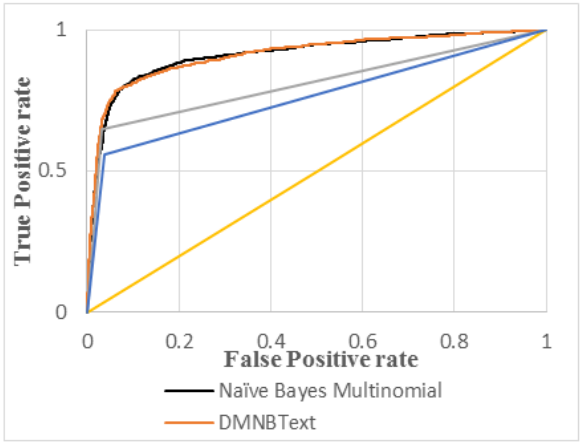

Fig.2. ROC AUC of algorithms

Finally, in order to consider one best algorithm for classification in SMS spam scenario, we performed a final test which is based on the error rate of both the top performing algorithms when data is reduced. In this experiment, percentage removal is used along with Naïve Bayes Multinomial and DMNBText classifiers. The experiment is performed in a way that from 10% to 90% data is removed from the dataset and rest is fed to the classifier and results are noted accordingly. Form this experiment a clear view is obtained that which algorithm performs how well on less data and is able to adapt to more data as well. Cross validation is used in this experiment which runs from 1 to 10 times on the dataset. Error rate is shown in Fig. 3. From Fig 3 it can be seen that error rate of Naïve Bayes Multinomial is less even when 90% of data is removed from the dataset but DMNBText have more error rate when there is less data. So Naïve Bayes Multinomial is

best algorithm as far as time of classification, error rate and number of false positives are concerned.

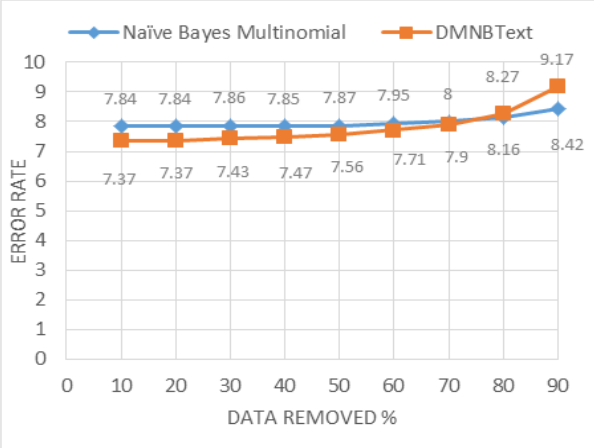

Fig.3: Comparison of Error Rate of NB Multinomial and DMNBText by removing percentage of data

V. CONCLUSION AND FUTURE WORK

There are number of algorithms and techniques available for spam filtering, few of them are more suitable on shorter messages According to the literature review that has been performed in this paper, Bayesian is the simplest and easy to implement technique while simple SVM is slower on larger datasets and also does not performs optimally in case of large number of attributes. Both the DMNBText and Naïve Bayes Multinomial performed well but the later had an upper hand in case of the false positives which were less as compared to the former. SMO and Liblinear showed good accuracy but they also took larger time for building model which is not feasible in smartphone's environment. For features, bag of word approach was used along with frequency of words. In future, other techniques such as n-grams and lexical profiles [24] can also be explored to find their suitability for spam detection of small text. These techniques have potential to produce better results in case of languages such as Roman Urdu which often exhibit varied spellings due to their dependency on phonetics.

VI. REFERENCES

1. F. T. Commission, "Unsolicited Commercial E-Mail," Subcommittee on Telecommunications, Trade and Consumer Protection of the Committee on Commerce, US House of Representatives, 3 Nov. 1999.
2. "U.S. Mobile Messaging 2011-2015 Forecast – The Evolving Role of SMS and MMS," Market Research Tech Rep, 2015. [Online]. Available: http://www.marketresearch.com/IDC-v2477/Mobile-Messaging-Forecast-Evolving-Role-6276827/. [Accessed June 2015].
3. PTA, "Pakistan Telecommunication Authority," [Online]. Available: http://www.pta.org.pk.
4. Javed and H. Afzal, "Opinion analysis of Bi-lingual Event Data from Social Networks," in ESSEM@AI*IA, Italy, 2013.
5. Javed and H. Afzal, "Creation of Bi-lingual Social Network Dataset using Classifiers," in Machine Learning and Data Mining in Pattern Recognition, St Petersburg, Springer International Publishing, 2014, pp. 523-533.
6. Javed, H. Afzal, A. Majeed and B. Khan, "Towards Creation of Linguistic Resources for Bilingual Sentiment Analysis of Twitter Data," in Natural Language Processing and Information Systems:19th International Conference on Applications of Natural Language to Information Systems, NLDB 2014, Montpellier, Springer International Publishing, 2014, pp. 232-236.
7. H. Afzal and K. Mehmood, "Spam Filtering of Bi-Lingual Tweets Using Machine Learning," in 18th International Conference on Advanced Communication Technologies (ICACT 2016), Korea, 2016.
8. Irvine, J. Weese and C. Callison-Burch, "Processing informal, Romanized Pakistani text messages," in Proceedings of the Second Workshop on Language in Social Media. Association for Computational Linguistics, Stroudsburg, 2012.
9. D. Longzhen, L. An and H. Longjun, "A New Spam Short Message Classification," in First International Workshop on Education Technology and Computer Science, vol.2, no., pp.168,171, 7-8 March 2009.
10. H.-y. Zhang and W. Wang, "Lazy Associative Classification for Content-based Spam Detection," in Web Congress LA-Web. Fourth Latin American , vol., no., pp.154,161, Oct 2006.
11. H.-y. &. W. W. Zhang, "Application of Bayesian method to spam sms filtering," in International Conference on Information Engineering and Computer Science, 1-3., 2009.
12. K. Mathew and B. Issac, "Intelligent spam classification for mobile text message," in International Conference on Computer Science and Network Technology (ICCSNT), vol.1, no., pp.101,105, 24-26, Dec. 2011.
13. M. L. Group, "Data Mining Software in Java," University of Waikato, [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/.
14. J. M. G. Hidalgo, G. C. Bringas, E. P. Sánz and F. C. García, "Content based SMS spam filtering.," in Proceedings of the 2006 ACM symposium on Document engineering (DocEng '06). ACM, 107, New York, NY, USA, 2006.
15. Y. Yang, "An evaluation of statistical approaches to text categorization.," in Information Retrieval, 1(1/2):69-90., 1999.
16. Y. Yang and J. Pedersen., "A comparative study on feature selection in text categorization," in Proceedings of the 14th International Conference on Machine Learning., 1997.
17. J. Quinlan, Programs for Machine Learning, Morgan Kaufmann, 1993.
18. E. Frank and I. Witten., "Generating accurate rule sets without global optimization," in Machine Learning: Proceedings of the Fifteenth International Conference., 1998.
19. V. V. a. D. W. H. Drucker, "Support vector machines for spam categorization.," in IEEE Transactions on Neural Networks, 10(5):1048–1054, 1999.
20. T. A. Almeida, J. M. G. Hidalgo and A. Yamakami, "Contributions to the study of SMS spam filtering: new collection and results," in Proceedings of the 11th ACM symposium on Document engineering (DocEng '11). ACM, New York, NY, USA, 25, 2011.
21. O. Lund, "Methods Applied in Immunological Bioinformatics," in Performance Measures for Prediction Methods, Cambridge, Massachusetts, The MIT Press, pp. 99-101.
22. J. Su, H. Zhang, C. X. Ling and S. Matwin, "Discriminative Parameter Learning for Bayesian Networks," in Proceedings of the

25th International Conference on Machine Learning, Helsinki, Finland, ACM, 2008, pp. 1016--1023.

23. M. L. Group, "LIBLINEAR -- A Library for Large Linear Classification," National Taiwan University, [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/liblinear/.
24. H. Afzal, R. Stevens and G. Nenadic, "Towards semantic annotation of bioinformatics services: building a controlled vocabulary," in Third International Symposium on Semantic Mining in Biomedicine., Turku, 2008.