VOLUME 6 NUMBER 2 December 2017
ISSN 2304-7712 (Print)
ISSN 2304-7720 (Online)

International Journal of Advanced Engineering and Science

ELITE HALL PUBLISHING HOUSE

ELITE HALL PUBLISHING HOUSE

International Journal of Advanced Engineering and Science

ABOUT JOURNAL

The International Journal of Advanced Engineering and Science (Int. j. adv. eng. sci. / IJAES) was first published in 2012, and is published semi-annually (May and November). IJAES is indexed and abstracted in: **ProQuest, Ulrich's Periodicals Directory, EBSCO Open Access Journals, Scientific Indexing Service, getCITED, ResearchBib, IndexCopernicus, NewJour, Electronic Journals Library, Directory of Research Journals Indexing, Open J-Gate, CiteFactor, JournalSeek, WZB Berlin Social Science Center, GEOMAR Library Ocean Research Information Access**. Since 2013, the IJAES has been included into the ProQuest one of the leading full-text databases around the world.

The International Journal of Advanced Engineering and Science is an open access peer-reviewed international journal for scientists and engineers involved in research to publish high quality and refereed papers. Papers reporting original research or extended versions of already published conference/journal papers are all welcome. Papers for publication are selected through peer review to ensure originality, relevance, and readability.

ISSN 2304-7712

International Journal of Advanced Engineering and Science

Publisher: Elite Hall Publishing House

Editor in Chief:

Editorial Board:

Web: http://ijaes.elitehall.com/
ISSN 2304-7712 (Print)
ISSN 2304-7720 (Online)

SPAM E-MAIL CHARACTERIZATION: AN EXPERIMENTAL PERFORMANCE COMPARISON OF MACHINE LEARNING

Avijit Mallik[1], Md. Sabbir Ahmad[2a], Md. Arman Arefin[1], Md. Sarwar Hosen[3]

[1]Dept. of Mechanical Engineering, Rajshahi University of Engineering & Technology, Rajshahi-6204, Bangladesh

[2*]Dept. of Computer Science & Engineering, Rajshahi University of Engineering & Technology, Rajshahi-6204, Bangladesh

[3] Dept. of Civil Engineering, Rajshahi University of Engineering & Technology, Rajshahi-6204, Bangladesh

[a]CEO, Appsplorer Technologies, Mirpur-DOHS, Dhaka-1216, Bangladesh

Email: avijitme13@gmail.com, contact.jim13@gmail.com* , dipto70@yahoo.com , sarwarce13@gmail.com

(All authors have contributed equally in this research)

Abstract

The increasing volume of unsolicited mass e-mail (otherwise called spam) has generated a need for reliable against spam filters. Utilizing a classifier based on machine learning techniques to naturally filter out spam e-mail has drawn many researchers' attention. In this paper, we review some of relevant ideas and do a set of systematic experiments on e-mail categorization, which has been conducted with four machine learning calculations applied to different parts of e-mail. Experimental results reveal that the header of e-mail provides very useful data for all the machine learning calculations considered to detect spam e-mail.

Keywords

E-mail, Spam, Machine Learning, Networking.

1. Introduction

In recent years, Internet e-mails have become a typical and imperative medium of correspondence for nearly everyone. However, spam, otherwise called unsolicited commercial/mass e-mail, is a bane of e-mail correspondence. There are numerous serious problems associated with developing volumes of spam. Spam is not just a waste of storage space and correspondence data transfer capacity, yet additionally a waste of time to tackle. Several arrangements have been proposed to overcome the spam problem. Among the proposed methods, much interest has focused on the machine learning techniques in spam filtering. They include rule learning [3], Naïve Bayes [1, 6], decision trees [2], bolster vector machines [4] or blends of different learners [7]. The fundamental and normal concept of these approaches is that utilizing a classifier to filter out spam and the classifier is learned from preparing information rather than constructed by hand. Therefore, it can result in better performance [9].

From the machine learning viewpoint, spam filtering based on the textual content of e-mail can be viewed as a special case of text categorization, with the categories being spam or non-spam [5]. Sahami et al. [6] employed Bayesian order technique to filter garbage e-mails. By making use of the extensible framework of Bayesian modeling, they cannot just employ customary

ISSN 2304-7712

document order techniques based on the text of e-mail, however they can likewise easily incorporate area knowledge to go for filtering spam e-mails.

Drucker et al. [4] used help vector machine (SVM) for arranging e-mails as indicated by their contents and compared its performance with Ripper, Rocchio, and boosting decision trees. They concluded that boosting trees and SVM had acceptable test performance in terms of precision and speed. However, the preparation time of boosting trees is inordinately long. Androutsopoulos et al. [1] extended the Naïve Bayes (NB) filter proposed by Sahami et al. [6], by investigating the effect of different number of features and preparing set sizes on the filter's performance. Meanwhile, they compared the performance of NB to a memory-based approach, and they discovered both above mentioned methods clearly outperform a common keyword-based filter.

The objective of this paper is to evaluate four respective machine learning calculations for spam email categorization. These techniques are Naïve Bayes (NB), term frequency – inverse document frequency (TF-IDF), K-nearest neighbor (K-NN), and bolster vector machines (SVMs). Also, we examine different parts of an e-mail that can be exploited to improve the categorization ability. We considered the accompanying five blends of an e-mail message: all (A), header (H), Body (B), subject (S), and body with subject (B+S). The above-mentioned four methods with these features are compared to help evaluate the relative merits of these calculations, and suggest directions for future works. The rest of this paper is organized as takes after.

2. Machine learning strategies and highlights in the email

In this area, we audit the machine learning calculations in the writing that utilized for email order (or against spam sifting). They incorporate Naïve Bayes (NB), term recurrence – opposite report recurrence (TF-IDF), K-closest neighbor (K-NN), and bolster vector machines (SVMs).

2.1 Term recurrence backwards record recurrence

The regularly embraced portrayal of an arrangement of messages is as term weight vectors which utilized as a part of the Text Processing Model [8]. The term weights are genuine numbers demonstrating the methodicalness of terms in recognizing a report. In light of this idea, the heaviness of a term in an email message can be registered by the *tf.idf*. The *tf* (term recurrence) shows the quantity of times that a term t shows up in an email. The *idf* (backwards report recurrence) is the reverse of record recurrence in the arrangement of messages that contain *t*. The *tf.idf* weighting plan is characterized as:

$$w_{ij} = tf_{ij} . \log(\frac{N}{df_i}) \qquad (1)$$

Where w_{ij} is the heaviness of *i*-th term in the *j*-th email, tf_{ij} is the quantity of times that *i*-th term happens in the *j*-th email, N is the aggregate number of messages in the gathering, and df_i is the quantity of messages in which the *i*-th term happens.

2.2 K-nearest neighbor

The most basic instance-based method is the K-nearest neighbor (K-NN) algorithm. It is an exceptionally simple method to classify documents and to show great performance on content categorization tasks [10]. On the off chance that we want to apply K-NN method to classify messages, the emails of the training set have to be recorded and then change over them into a

ISSN 2304-7712

report vector representation. When classifying another email, the similarity between its record vector and each one in the training set has to be registered. Then, the categories of the k nearest neighbors are resolved and the category which occurs most as often as possible is chosen.

2.3 Naïve Bayes

The Naïve Bayes (NB) classifier is likelihood based approach. The fundamental idea of it is to discover whether an email is spam or not by taking a gander at which words are found in the message and which words are missing from it. In the writing, the NB classifier for spam is characterized as takes after:

$$C_{NB} = \arg amix\ P(c_i) \prod_k \mathrm{P}(\mathrm{w}_k|\mathrm{c}_i),\ c_i \epsilon T \qquad (2)$$

Where T is the arrangement of target classes (spam or non-spam), and P($w_k|c_i$) is the likelihood that word w_k happens in the email, given the email has a place with class c_i. The probability term is assessed as:

$$P(w_k|c_i) = \frac{n_k}{N} \qquad (3)$$

Where n_k is the quantity of times word w_k happens in messages with class ci, and N is the quantity of words in messages with class ci.

2.4 Support vector machine

Support vector machine (SVM) has become exceptionally popular in the machine learning group because of its great generalization performance and its ability to handle high-dimensional data by using kernels.

According the description given in Heckerman et al. [11], an email may be represented by a feature vector x that is composed of the various words from a dictionary framed by analyzing the gathered messages. Thus, an email is classified as spam or non-spam by playing out a simple spot item between the features of an email and the SVM display weight vector,

$$y = w.x - b \qquad (4)$$

Where y is the result of classification, w is weight vector corresponding to those in the feature vector x, and b is the bias parameter in the SVM show that controlled by the training process.

2.5 The structure of an email

In addition to the instant message of an email, an email has additional information in the header. The header contains many fields, for example, trace information about which a message has passed (Received :), where the sender wants replies to go (Reply-To :), one of a kind of ID of this (Message-ID :), format of substance (Content-Type :), and so on. **Figures 1** illustrates the header of an email. Besides comparing the categorization performance among the learning algorithms, we planned to make sense of which parts of an email have critical impact on the classification results.

Therefore, five features of an email: all (A), header (H), body (B), subject (S), and body with subject (B+S), are used to

ISSN 2304-7712

evaluate the performance of four machine learning algorithms. Furthermore, we also considered four cases that whether stemming or stopping methodology was applied or not.

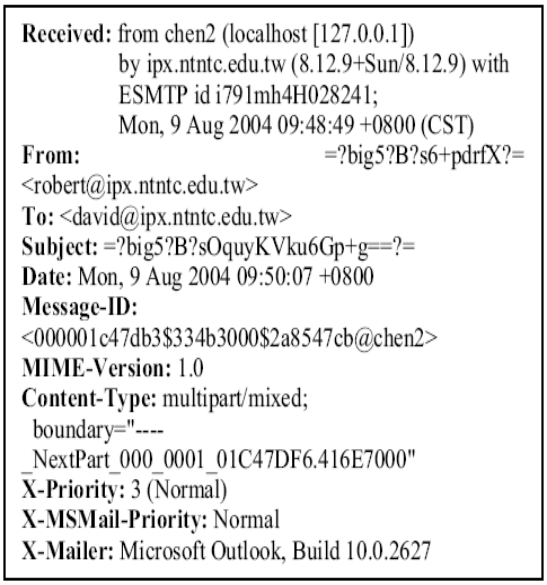

Figure 1: Header of an e-mail

3. Experimental outcomes and discussions

With a specific end goal to test the execution of previously mentioned four techniques, two corpora were utilized. The main (Corpus I) comprises of our messages over a current three – month time span. For tests, we erased a few messages whose messages were too short or did not contain any substance and acquire 1050 spam and 1057 non-spam. The second (Corpus II) we received is accessible at www.spamassassin.org. This file contains 2100 spam and 2107 non-spam messages. Trials were kept running with various preparing and test sets. The principal match of preparing and test set is made by part every corpus at a proportion of 20:80. The second match and the third one are 30:70 and 40:60, separately.

In email grouping errands, the execution is often measured as far as precision. Give N_{legit} and N_{spam} a chance to mean the aggregate quantities of non-spam and spam messages, individually, to be ordered by the machine learning strategy, and n(C-->V) the quantity of messages having a place with classification C that the technique delegated class V (here, C, Vϵ{legit, spam}). The precision is characterized as following equation:

$$Accuracy = \frac{number\ of\ e-mails\ correctly\ categorized}{total\ number\ of\ e-mails} = \frac{n(legit \rightarrow legit)+n(spam \rightarrow spam)}{N_{legit}+N_{spam}} \quad (5)$$

The general exhibitions of considered learning calculations in various investigations are appeared in Tables 1 and 2. From the outcomes, we found the accompanying wonders.

ISSN 2304-7712

1. Great execution of NB, TF-IDF and SVM with header data. NB and TF-IDF performed sensibly reliable and great in various exploratory settings. While SVM performed well with the exception of the component of subject. It appears that the subject is insufficient for high exactness characterization in SVM.

2. Poor execution of KNN technique. KNN played out the most exceedingly awful among all considered techniques and the poorest in all cases. Notwithstanding, if the more pre-handling errands are used (i.e., stemming and ceasing are connected together), the better KNN performs.

3. No impact of stemming; however ceasing can improve the email grouping. Stemming did not make any huge change for all calculations in execution; however it diminished the measure of the list of capabilities. Then again, when the ceasing methodology is utilized, that is, disregarding a few words that don't convey significance in regular dialect, we can improve execution. The wonder is evident particularly in K-NN technique as appeared in **Figure 2**.

4. Great execution with header. Among four machine learning calculations, the execution with header was the best. This implies much data can be gotten from the header and after that the fields in the header can go for arranging messages accurately.

5. Poor execution with subject or body. The poor execution of every calculation happens in subject or body. The reasons might be that the previous gives too minimal valuable data. Actually, the last contains excessively pointless data to order messages.
From the perception, we realize that albeit some learning calculations can accomplish acceptable outcomes, we may endeavor to enhance the come about by brushing some of them. Here, we coordinate TF-IDF and NB techniques and apply them to two corpora. The trial comes about are appeared in the last section of **Tables 1** and **2**. They demonstrate that the precision can be enhanced by the new mixture approach.

Table 1: Performance of four machine learning algorithms in **CORPUS-I**

Features	Preprocessing		Algorithms				
	stemming	stopping	NB	TFIDF	K-NN	SVM	TFIDF+NB
A			87.60	88.37	49.20	91.11	90.42
A		OK	88.32	90.19	87.72	92.26	91.46
A	OK		87.57	88.14	51.31	91.29	90.64
A	OK	OK	88.71	90.08	88.99	92.15	90.97
H			93.36	94.25	70.38	92.99	93.87
H		OK	93.21	95.29	87.71	92.95	95.30
H	OK		93.23	94.70	73.02	92.87	93.59
H	OK	OK	93.46	95.24	87.58	92.86	94.90
B			87.46	89.31	47.50	83.34	92.04
B		OK	88.74	90.08	81.65	85.79	90.69
B	OK		85.78	89.41	46.80	83.53	91.66
B	OK	OK	89.47	90.19	81.97	85.84	90.39

ISSN 2304-7712

S			83.71	83.35	62.55	77.29	84.92
S		OK	83.85	88.17	82.54	74.74	86.35
S	OK		83.60	93.61	67.19	77.97	85.45
S	OK	OK	84.04	87.64	82.64	74.17	84.04
B+S			87.22	86.64	78.40	84.71	88.48
B+S		OK	87.81	88.88	76.58	87.20	89.83
B+S	OK		87.62	87.48	48.92	85.17	89.36
B+S	OK	OK	88.84	88.90	79.49	87.17	90.68
Average			88.18	89.50	70.10	86.27	90.25

Table 2: Performance of four machine learning algorithms in **CORPUS-II**

Features	**Preprocessing**		**Algorithms**				
	stemming	**stopping**	**NB**	**TFIDF**	**K-NN**	**SVM**	**TFIDF+NB**
A			87.56	89.49	48.57	91.11	91.71
A		OK	88.73	90.30	88.64	91.58	91.78
A	OK		87.56	88.52	48.71	91.13	92.46
A	OK	OK	88.76	90.08	89.99	92.28	92.72
H			93.18	91.95	69.75	93.00	95.14
H		OK	93.22	91.40	89.21	92.89	91.61
H	OK		93.23	90.86	73.15	92.59	92.01
H	OK	OK	91.02	89.89	88.48	92.71	92.64
B			84.33	79.29	46.98	89.82	88.06
B		OK	89.29	84.10	82.23	89.41	89.71
B	OK		83.31	87.47	46.49	85.41	89.91
B	OK	OK	89.18	84.08	83.11	89.10	90.19
S			83.40	82.84	65.36	77.29	85.69
S		OK	83.82	87.84	81.89	74.78	84.81
S	OK		84.02	83.74	61.68	77.97	87.62
S	OK	OK	84.04	87.49	82.03	74.10	87.88
B+S			86.49	84.58	47.15	85.51	88.46
B+S		OK	87.51	88.52	79.34	87.00	89.81
B+S	OK		85.43	84.57	47.03	84.80	88.55
B+S	OK	OK	88.77	85.50	81.06	87.21	89.53
Average			87.64	87.13	70.04	86.98	90.01

ISSN 2304-7712

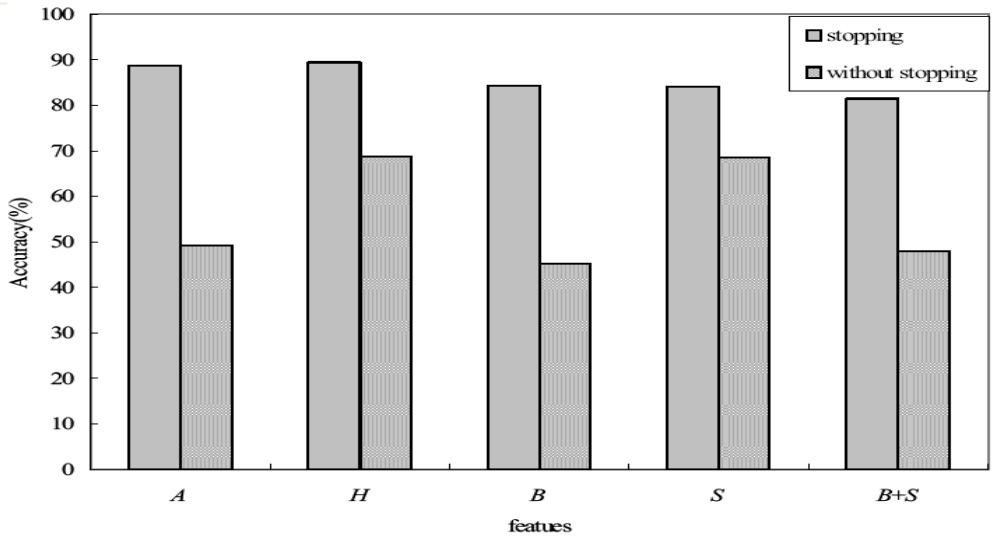

Figure 2: Performance of K-NN method in all features in **CORPUS-I**

4. Conclusion

The location of spam email is an imperative issue of data advancements, and machine learning has a focal part to play in this subject. In this paper, we exhibited an experimental assessment of four machine learning calculations for spam email arrangement. These methodologies NB, TF-IDF, K-NN, and SVM, were connected to various parts of an email keeping in mind the end goal to think about their execution. Test comes about demonstrate that NB, TF-IDF, and SVM yield preferred execution over K-NN. The wonder likewise found, at any rate with our test corpora, that arrangement with the header was the most exact than different parts of an email. Then again, we attempt to join two techniques (TF-IDF and NB) to accomplish the most right arrangement. It was discovered that incorporating diverse learning calculations really is by all accounts a promising way.

Acknowledgement

This work was partially supported by Appsplorer Technologies, Bangladesh and Dept. of Mechanical Engineering, Rajshahi University of Engineering & Technology, Bangladesh. Comments from anonymous referees are highly appreciated.

References:

[1] Androutsopoulos, Ion, et al. "Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach." *arXiv preprint cs/0009009*(2000).
[2] Pang, Xiu-Li, Yu-Qiang Feng, and Wei Jiang. "A spam filter approach with the improved machine learning technology." *Natural Computation, 2007. ICNC 2007. Third International Conference on*. Vol. 2. IEEE, 2007.
[3] Cohen, William W. "Learning rules that classify e-mail." *AAAI spring symposium on machine learning in information access*. Vol. 18. 1996.
[4] Drucker, Harris, Donghui Wu, and Vladimir N. Vapnik. "Support vector machines for spam categorization." *IEEE Transactions on Neural networks*10.5 (1999): 1048-1054.
[5] Blanzieri, Enrico, and Anton Bryl. "A survey of learning-based techniques of email spam filtering." *Artificial Intelligence Review* 29.1 (2008): 63-92.
[6] Sahami, Mehran, et al. "A Bayesian approach to filtering junk e-mail." *Learning for Text Categorization: Papers from the 1998 workshop*. Vol. 62. 1998.
[7] Sakkis, Georgios, et al. "Stacking classifiers for anti-spam filtering of e-mail." *arXiv preprint cs/0106040* (2001).

ISSN 2304-7712

[8] Vogel, Claude. "Text processing and retrieval system and method." U.S. Patent No. 5,963,965. 5 Oct. 1999.
[9] Schneider, Karl-Michael. "A comparison of event models for Naive Bayes anti-spam e-mail filtering." *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003.
[10] Yang, Yiming. "An evaluation of statistical approaches to text categorization." *Information retrieval* 1.1 (1999): 69-90.
[11] Heckerman, David, et al. "Method and system for identifying junk e-mail." U.S. Patent Application No. 10/278,591.

ISSN 2304-7712

International Journal of Advanced Engineering and Science
http://ijaes.elitehall.com/