Fjrs.futa.edu.ng

FUTA Journal of Research in Sciences

ISSN: 2315 – 8239 (Print); E-ISSN: 2489 - 0413

FUTA Journal of Research in Sciences, Vol. 13 (1) 2017:177-185

SMS SPAM FILTERING FOR MODERN MOBILE DEVICES

***N.A Azeez and O. Mbaike**
Department of Computer Sciences, University of Lagos, Nigeria.

E - mails: nazeez@unilag.edu.ng, onyekachi.mbaike@gmail.com

ABSTRACT
This work examines solutions to the growing problem of spam and fraudulent messages that are prevalent in the mobile phone industry today. It begins with an examination of some common methods for detecting spam messages such as: the Rule-based method and the Statistical Learning method using Naive Bayes approach. This work specifically explores Naive-Bayes classifier for categorizing messages based on their resemblance with words that feature in other spam and non-spam messages in the training set, thereby reducing the number of spams that get through to the end user and completely eliminate false positives (messages that are misclassified as spam). Incorporated in the dataset for this project is the SMS Spam Corpus v.0.1 Big. It has 1,002 SMS ham (legitimate) messages and 322 spam messages. For the initial training and testing of the Naive-Bayes classifier, Python 2.7 interpreter, Sublime Text text-editor, Plotly for data visualization and comparing results were used while Java Development Kit, Android SDK, Android Studio and Android Emulator were used for deployment. It can be concluded that using a spam threshold of 0.7 along with adjustments to the Naive Bayes algorithm, we obtained some desirable results. In an attempt to improve on the method used in this work, we are currently working on how to use hybridized machine learning algorithms for detecting Spam messages in mobile devices.

Keywords***:*** SMS; Spam filtering; Naive Bayesian filter; threshold

INTRODUCTION
Spam has been a large problem on the internet for as long as e-mail and personal computers have been ubiquitous. As a result, numerous methods have been proposed to reduce the ease at which spammers can get messages to their targets. These efforts to fight spam on the internet have not totally eradicated it yet, but have made it increasingly difficult (and a lot less lucrative) for those in the business of email spamming (Nureni and Irwin, 2010).
In recent years, there has been tremendous growth in the mobile phone industry globally, replacing personal computers and fixed phones. This new, large market, coupled with the decreasing charges for SMS (especially in bulk) worldwide presents a great opportunity for spammers and fraudsters alike to reach their unsuspecting targets easily, and at little cost without the resistance they were met with on the web. As a result of this, SMS spam has become very popular in many parts of the world (Azeez *et al.*, 2011).

A number of people are involved in generating spam messages ranging from legitimate businesses and organizations trying to market their products and services to fraudulent individuals who aim to deceive people to make a profit. Many network providers are also involved in generating these unsolicited, marketing messages (Thomason, 2007).

Some factors that differentiate spam SMS from their email counterparts and make them a lot more difficult to detect are: Their short length (usually 140 - 160 characters), absence of headers, really small number of test data sets available for SMS and widespread use of abbreviations within text content (Azeez and Ademolu, 2016).

RULE-BASED SPAM FILTERING

Rule-based filtering techniques rely on the specification of lists of words or regular expressions disallowed in messages (Ayofe *et al.,* 2010). Thus, if a user receives spam advertising "herbal Viagra", the user might place this phrase in the filter configuration. The system would then reject any message containing the phrase. The user may also filter based on details of the sender of the message, either the displayed name or the number that comes with the message (Zdziarski, 2005).

STATISTICAL LEARNING FILTERING

Statistical (or Bayesian) filtering once set up, requires no administrative maintenance per se: instead, users mark messages as spam or non-spam and the filtering software learns from these judgments (Robinson, 2003). Thus, a statistical filter does not reflect the software author's or administrator's biases as to content, but rather the user's biases. For example, a biochemist who is researching Viagra won't have messages containing the word "Viagra" automatically flagged as spam, because "Viagra" will show up often in his or her legitimate messages. Still, spam emails containing the word "Viagra" do get filtered because the content of the rest of the spam messages differs significantly from the content of legitimate messages (Healy et. al., 2005). A statistical filter can also respond quickly to changes in spam content, without administrative intervention, as long as users consistently designate false negative messages as spam when received in their email. Statistical filters (Azeez and Lasisi, 2016) can also look at message senders, thereby considering not just the content but also peculiarities of the transport mechanism of the message (Sun, 2009).

Typical statistical filtering uses single words in the calculations to decide if a message should be classified as spam or not. A more powerful calculation can be made using groups of two or more words taken together. Then random "noise" words cannot be used as successfully to fool the filter (Wu *et al.,* 2008).

METHODOLOGY

The methodology adopted in this research is the Naïve Bayes method.

Naive Bayes

Naive Bayes is a stochastic technique that evaluates the similarities among objects by computing individual or partial probability and feed it into the conditional probabilities of items used for basis of similarities. Naive categorizes N number of feature vectors into n number of identified definite and unique clusters (Azeez and Babatope, 2016). The small n number of unique clusters may be part of the features or separated. They may be considered as seeded feature vector (Azeez and Venter, 2013).

The task of this algorithm is to compute the partial probabilities of individual vectors and then compute their conditional probabilities in relation with the seeded vectors. Finally, the cluster with the maximum conditional probability is selected as the most appropriate cluster for the vector (Azeez and Iliyas, 2016).

Naive Bayes Classification

Naive Bayes classification is based on Bayesian probability learning. It assumes conditional independence of attributes given the target value of the instance (Azeez *et al.,* 2015). For a learning task based on Naive Bayes classification, each instance in the dataset is represented as a conjunction of attribute values where the target function f(x) can take values from a finite set V. A new instance to be classified is presented in the form of a tuple (a1, a2, a3 … an) and the learner is supposed to classify this new instance (Teevan *et al.,* 2003).

The above probabilities are based on the frequencies of attribute values in the training data. In this case, there is no explicit search in the hypothesis space, just frequency counting. A more detailed explanation can be found in (Mitchell, 1997).

The steps taken for the implementation are as follows:

1. Data collection
2. Feature extraction
3. Training the Naive Bayes classifier
4. Classifying incoming messages as spam of non-spam

Naive Bayes Formula

The formula used in the software to calculate the probability of a message being spam based on a given word 'W':

$$\Pr(S/W) = \frac{\Pr(W/S) * \Pr(S)}{\Pr(W/S) * \Pr(S) + \Pr(W/H) \quad * \quad \Pr(H)} \quad (1)$$

Where:

- Pr(S|W) is the probability that a message is spam, knowing that the word 'W' is contained in it;
- Pr(S) is the overall probability that any given message is spam;
- Pr(W|S) is the probability that the word 'W' appears in spam messages;
- Pr(H) is the overall probability that any given message is not spam (i.e. is "ham");
- Pr(W|H) is the probability that the word 'W' appears in ham messages.

The formula used to combine the probability of a message being spam given the probabilities of several words with probabilities P1, P2, P3, …, Pn is given by:

$$\Pr(S/W) = \frac{P1 * P2 * \ldots\ldots\ldots\ldots * Pn}{P1 * P2 \ldots * Pn + (1-P1) * (1-P2) * \ldots * (1-Pn)} \quad (2)$$

This is the formula referenced by Paul Graham in his 2004 article (Graham, 2004).

DATASET EXTRACTION

The data set used for this project is the SMS Spam Collection v.1. It is a public set of SMS labeled messages that have been collected for mobile phone spam research. It has one collection composed by 5,574 English, real and non-encoded messages, tagged according being legitimate (ham) or spam (Murynets and Piqueras, 2012).

This corpus has been collected from free sources on the Internet:

1. Incorporated in the dataset for this project is the SMS Spam Corpus v.0.1 Big. It has 1,002 SMS ham(legitimate) messages and 322 spam messages and it is public available at:http://www.esp.uem.es/jmgomez/smsspamcorpus/.
2. We have also added a dataset provided by Opeyemi Obembe for use in spam filtering research provided here:http://obem.be/2014/09/07/building-an-sms-spam-filter.html

FEATURE EXTRACTION

Features were extracted by scanning the messages for alphanumeric characters, currency signs, dashes, apostrophes and currency signs. All other tokens considered as separators were ignored.

The number of times every token appears in each corpus was then computed. This step produced two tables, one for each corpus, mapping tokens to their frequency of occurrence.

After the data set was tokenized, experiments were then carried out to arrive at the optimal result (no false positives and a high percentage of correctly classified spam). These experiments were carried out using varying multiplier values with varying spamicity thresholds (the probability that an email is spam) in an attempt to slightly bias the probabilities and reduce false positives to 0.

While calculating the probability of a token being spam or ham (legitimate) using the Naïve Bayes formula, tokens that occurred in one corpus but not in the other were assigned a probability of 0.99.

DISCUSSION OF RESULTS

Given a base training data set, the application successfully detects spam sent to the user's device. Over time, any false positives that find their way through can be detected and used to train the application to detect messages like the one missed. This leads to more efficient spam detection over time.

Below we present some results considering sharp twists and fine adjustment to the typical Naive Bayes approach as proposed by (Graham, 2004) using our given data set of 1372 messages (370 spam and 1002 legitimate messages).The data set was split into two, half containing 1022 messages (802 legitimate messages and 220 spam) for training the classifier and the other containing 350 messages (200 legitimate messages and 150 spam) for testing the accuracy of the classifier.

The first major suggestion given in (Graham, 2004) is doubling the appearance of words classified as ham to bias the probabilities slightly and avoid false positives. Below are some tests run on the project data set based on this suggestion.

The second suggestion is that a threshold of 0.9 be set to determine if a message should be classified as spam or not. Below, the results for tests carried out using different values for both the multiplier and the thresholds are shown.
Data in *Table 1* shows that the optimal threshold value for determining if a message is spam is 0.7 with no false positives and a large number of correctly classified spam messages.

Table 2 shows that the optimal threshold value for determining if a message is spam given a multiplier of 2 is between 0.5 and 0.6 for this data set. It also shows that higher thresholds yield smaller numbers of false positives (which is highly desirable) and smaller number of correctly classified spam (which is less desirable).

It can be observed from *Table 3* that even though the number of false positives is greatly reduced by adding a multiplier value, the accuracy of the classifier in detecting spam is also greatly affected.

Table 1. Accuracy of NB classifier given varying spamicity threshold

Ham (200)		**Spam (150)**		**Total (350)**		**Threshold**
%	**correct**	**%**	**correct**	**%**	**correct**	
99	198	88.6667	133	94.5714	331	0.4
99.5	199	88.6667	133	94.8571	332	0.5
99.5	199	88.6667	133	94.8571	332	0.6
100	200	88.6667	133	95.1428	333	0.7
100	200	88	132	94.8571	332	0.8
100	200	88	132	94.8571	332	0.9

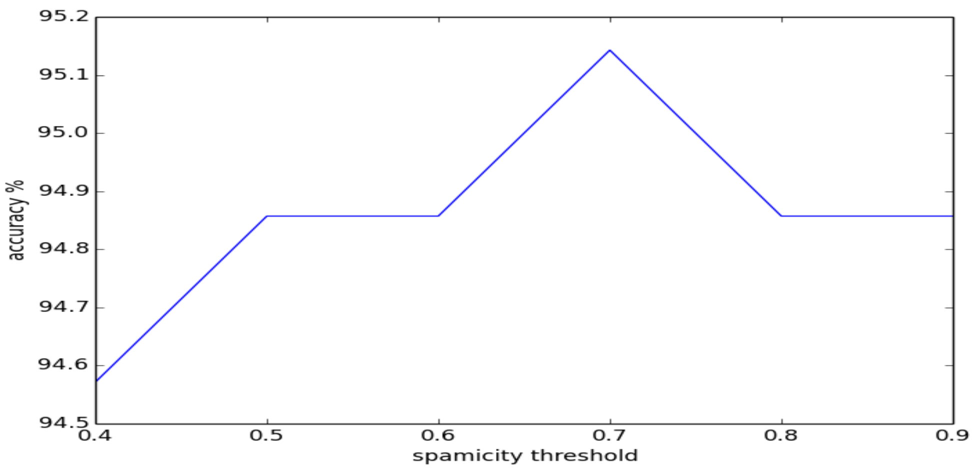


Fig. 1. Graphical representation of accuracy for different spasticity thresholds

Table 2. Accuracy of NB classifier given varying spamicity thresholds and a multiplier of 2

Ham (200)		**Spam (150)**		**Total (350)**		**Threshold**
%	**correct**	**%**	**correct**	**%**	**correct**	
99.5	199	87.3333	131	94.2857	330	0.4
100	200	87.3333	131	94.5714	331	0.5
100	200	87.3333	131	94.5714	331	0.6
100	200	86.6667	130	94.2857	330	0.7
100	200	85.3333	128	93.7142	328	0.8
100	200	84.6667	127	93.4285	327	0.9

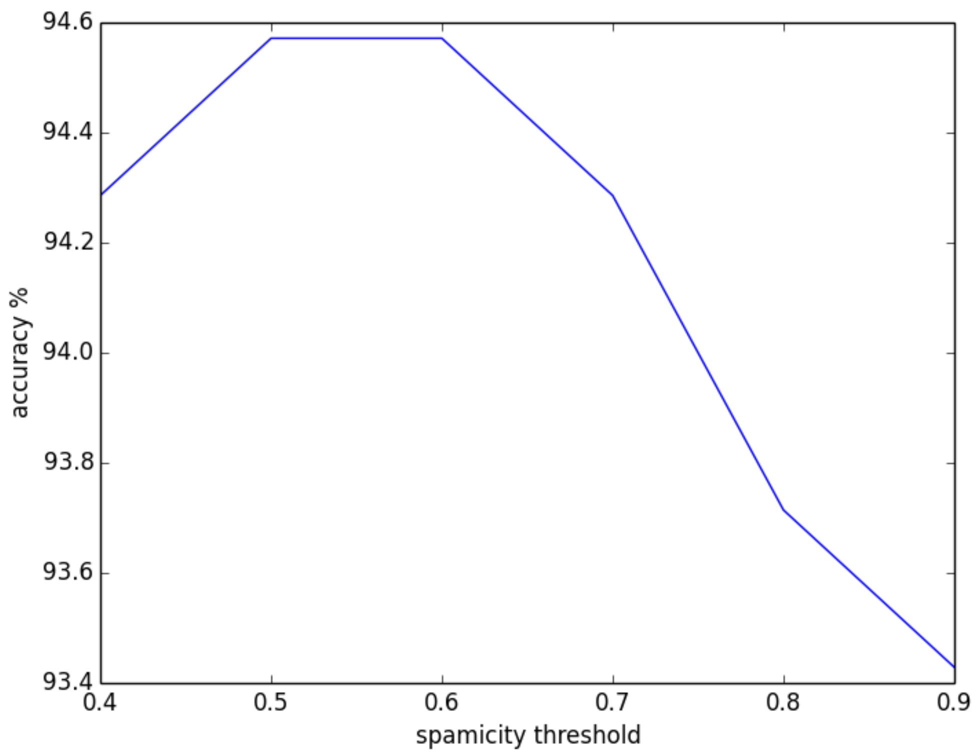


Fig. 2. Graphical representation of accuracy of NB classifier given varying spamicity thresholds and a multiplier of 2

Multiplier used in this case provides a better and greater final increase in the accuracy of Naïve Bayes classification and also assists in the readability of values obtained.

Table 3. Accuracy of NB classifier given varying values of the multiplier and a threshold of 0.9

Ham (200)		**Spam (150)**		**Total (350)**		**Multiplier**
%	**correct**	**%**	**correct**	**%**	**correct**	
99.5	200	88	131	94.29	332	1
100	200	84.67	131	93.43	327	2
100	200	84	131	93.14	326	3
100	200	82	130	92.29	325	4
100	200	82	128	92	323	5

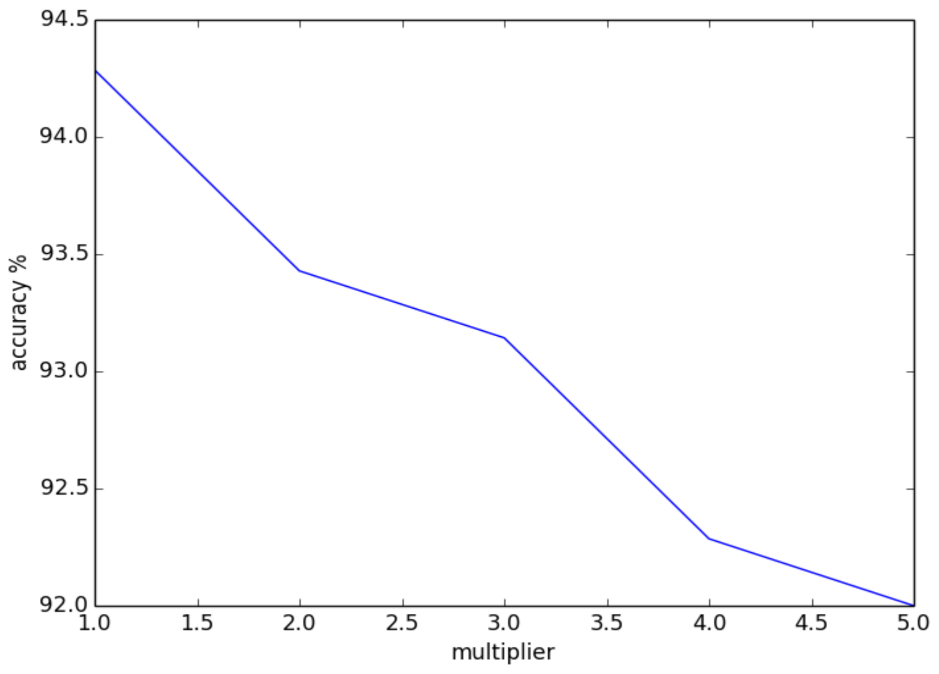
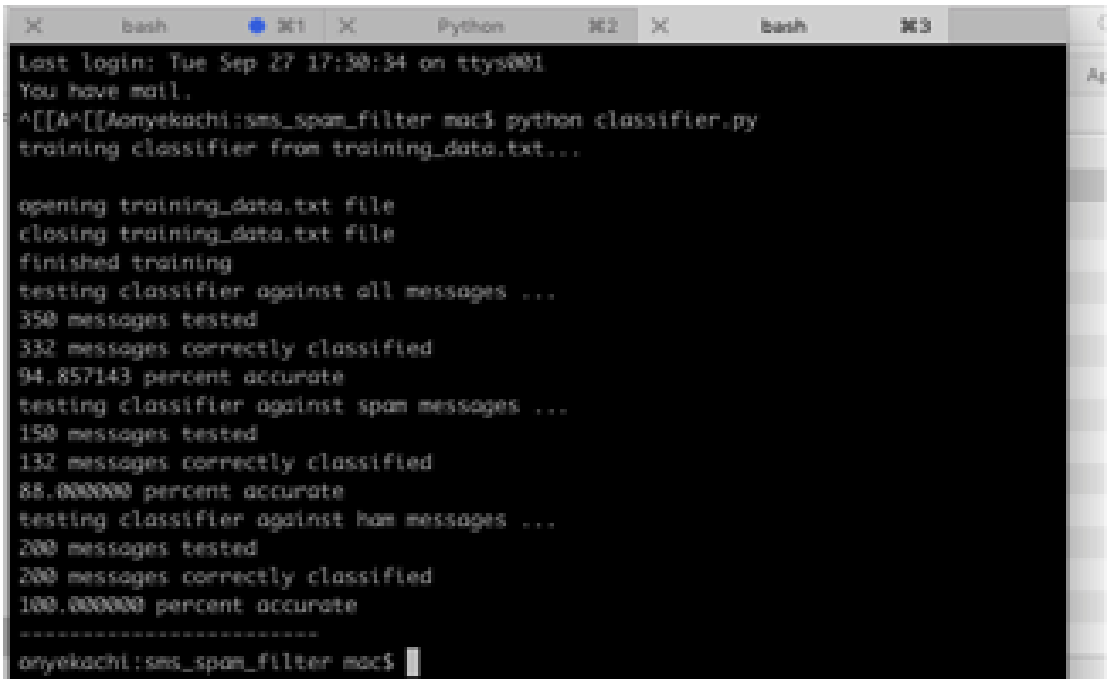


Fig. 3. Graphical representation of accuracy for different spasticity thresholds given a multiplier of 2



```
Last login: Tue Sep 27 17:30:34 on ttys001
You have mail.
^[[A^[[Aonyekachi:sms_spam_filter mac$ python classifier.py
training classifier from training_data.txt...

opening training_data.txt file
closing training_data.txt file
finished training
testing classifier against all messages ...
350 messages tested
332 messages correctly classified
94.857143 percent accurate
testing classifier against spam messages ...
150 messages tested
132 messages correctly classified
88.000000 percent accurate
testing classifier against ham messages ...
200 messages tested
200 messages correctly classified
100.000000 percent accurate
------------------------
onyekachi:sms_spam_filter mac$ 
```

Fig. 4. Initialization and running of the Naive Bayes classifier

DEVELOPMENT AND DEPLOYMENT

For the initial training and testing of the Naive-Bayes classifier, the tools used were: Python 2.7 interpreter, Sublime Text text-editor, Plotly for data visualization and comparing results. For the deployed application, the tools used were: Java Development Kit, Android SDK, Android Studio and Android Emulator. The application can be deployed on Android devices running OS version 4.2 and above. The classifier can however be reused and deployed in any python environment.

CONCLUSION

A spam filtering application for mobile devices using Naive Bayes algorithm was proposed and it correctly classified incoming messages received by users. The application categorizes messages based on their resemblance with words that feature in other spam and non-spam messages in the training set. Given the test data-set, it can be concluded that using a spam threshold of 0.7 along with adjustments to the Naive Bayes algorithm as proposed by Paul Graham returns the most desirable results. Further research is underway on how to optimize and leverage hybridized machine learning algorithms for detecting spam messages in mobile phones. This is necessary to obtain more efficient and dependable results.

REFERENCES

Ayofe, A.N, Adebayo, S.B, Ajetola, A.R, Abdulwahab, A.F (2010) "A framework for computer aided investigation of ATM fraud in Nigeria" International Journal of Soft Computing, Vol. 5, Issue 3 pp. 78-82

Azeez, N.A, Olayinka, A.F, Fasina, E.P, Venter, I.M. (2015) "Evaluation of a flexible column-based access control security model for medical-based information" Journal of Computer Science and Its Application. Vol. 22, Issue 1, Pages 14-25

Azeez, N. A., and Ademolu, O. (2016). CyberProtector: Identifying Compromised URLs in Electronic Mails with Bayesian Classification. 2016 International Conference Computational Science and Computational Intelligence (CSCI) (pp. 959-965). Las Vegas, NV, USA: IEEE.

Azeez, N. A., and Babatope, A. B. (2016). AANtID: an alternative approach to network intrusion detection. The Journal of Computer Science and its Applications. An International Journal of the Nigeria Computer Society, 129-143.

Azeez, N. A., and Iliyas, H. D. (2016). Implementation of a 4-tier cloud-based architecture for collaborative health care delivery. Nigerian Journal of Technological Development, 13 (1), 17-25.

Azeez, N. A., and Venter, I. M. (2013). Towards ensuring scalability, interoperability and efficient access control in a multi-domain grid-based environment. SAIEE Africa Research Journal , 104 (2), 54-68.

Azeez, N. A., Iyamu, T., and Venter, I. M. (2011). Grid security loopholes with proposed countermeasures. In E. Gelenbe, R. Lent, and G. Sakellari (Ed.), 26th International Symposium on Computer and Information Sciences (pp. 411-418). London: Springer.

Azeez, N.A., and Lasisi, A. A. (2016). Empirical and Statistical Evaluation of the Effectiveness of Four Lossless Data Compression Algorithms. Nigerian Journal of Technological Development, Vol. 13, NO. 2, December 2016, 64-73.

Graham, P. (2004). .*A plan for spam. In Reprinted in Paul Graham, Hackers and Painters, Big Ideas from the Computer Age,* O Really.

Healy, M., Delany, S., & Zamolotskikh, A. (2005). *An Assessment of CaseBased Reasoning for Short Text Message Classification*. In N. Creaney (Ed.), Proceedings of 16th Irish Conference on Artificial Intelligence and Cognitive Science, (AICS-05) (pp. 257–266).

Mitchell , T. M. (1997). Machine Learning . Ithaca, NY, United States of America: McGraw-Hill Science.

Murynets, A, and Piqueras, J (2012). Crime Scene Investigation: SMS Spam Data Analysis. In Proceedings of the Internet Measurement Conference, IMC'12, pages 441-452, Boston, MA, November 2012.

Nureni, A. A., and Irwin, B. (2010). Cyber security: Challenges and the way forward. Computer Science & Telecommunications, 29, 56-69.

Robinson, G (2003). *A statistical approach to the spam problem,* March 2003, pp. 107.

Sun, T. (2009). Spam Filtering based on Naïve Bayes Classification. Babeş-Bolyai University, Mathematics and Computer Science. Cluj-Napoca: Babeş-Bolyai University.

Teevan, J, Rennie, J, Shih, L & Karger, D (2003). *Tackling the poor assumptions of naive bayes text classifiers*. In ICML.

Thomason, A. (2007). *Blog spam: A review. In Conference on Email and Anti-Spam,* CEAS '07, Mountain View, CA, USA.

Wu, N., Wu, M., & Chen, S. (2008). *Real-time monitoring and filtering system for mobile SMS*. In Proceedings of 3rd IEEE Conference on Industrial Electronics and Applications (pp. 1319 –1324).

Zdziarski, J. (2005). Ending spam: Bayesian content filtering and the art of statistical language classification. No Starch Press, San Francisco.