Spam E-Mail Classification by Utilizing N-Gram Features of Hyperlink Texts

A. Selman Bozkir[1], Esra Sahin[2], Murat Aydos[3],
Ebru Akcapinar Sezer[4]
Hacettepe University Department of Computer Engineering
Ankara, Turkey
selman@cs.hacettepe.edu.tr, esrasahince@gmail.com,
maydos@hacettepe.edu.tr, ebru@hacettepe.edu.tr

Fatih Orhan[5]

COMODO Group
New Jersey, USA
fatih.orhan@comodo.com

Abstract— **With the advent of the Internet and reduction of the costs in digital communication, spam has become a key problem in several types of media (i.e. email, social media and micro blog). Further, in recent years, email spamming in particular has been subjected to an exponentially growing threat which affects both individuals and business world. Hence, a large number of studies have been proposed in order to combat with spam emails. In this study, instead of subject or body components of emails, pure use of hyperlink texts along with word level n-gram indexing schema is proposed for the first time in order to generate features to be employed in a spam/ham email classifier. Since the length of link texts in e-mails does not exceed sentence level, we have limited the n-gram indexing up to trigram schema. Throughout the study, provided by COMODO Inc, a novel large scale dataset covering 50.000 link texts belonging to spam and ham emails has been used for feature extraction and performance evaluation. In order to generate the required vocabularies; unigrams, bigrams and trigrams models have been generated. Next, including one active learner, three different machine learning methods (Support Vector Machines, SVM-Pegasos and Naive Bayes) have been employed to classify each link. According to the results of the experiments, classification using trigram based bag-of-words representation reaches up to 98,75% accuracy which outperforms unigram and bigram schemas. Apart from having high accuracy, the proposed approach also preserves privacy of the customers since it does not require any kind of analysis on body contents of e-mails.**

Index Terms— **Spam Email, Machine Learning, Active Learning, N-Grams, Bag of Words**

I. Introduction

Spam has been one of the leading problems in digital communication over the couple of decades. As reported in [1], the most common form of spam is email spamming which refers to receiving unwanted bulk email messages. Due to the very limited cost and great easiness in reaching millions of Internet users, spam has been excessively used by amateur advertisers and direct marketers [10]. On the other hand, spam mails do not only waste the time of Internet users but also causes several harms such as excessive bandwidth usage and potential virus attacks. According to the International Telecommunication Union, 70% of the emails around the world have been sent for spamming purpose [2].

In order to thwart the problem of spam emails, various approaches and methods have been proposed. Tretyakov [3] and Bhowmick & Hazarika [4] have collected these approaches under two headings: (1) *knowledge engineering (KE)* and (2) *machine learning (ML)*. While the former case comprises the methods of heuristic filters, blacklisting, whitelisting, greylisting, collaborative spam filtering, honey pots and signature schemes, latter one deals with classification of spam emails by employing machine learning methods over various extracted features from either content or header sections of emails. Though, they have satisfactory success in spam mail filtering, knowledge engineering based approaches are subjected to some problems and shortcomings. For instance, clear domains can be blocked because of spammer exploits or accounts of innocent users can be attacked by hackers [4]. Moreover, rules in those approaches should be updated frequently in order to adapt new types of threats. Apart from rule based heuristic techniques, ML based approaches achieve more success in filtering/classification results since they do not require any explicit rulesets. Instead, ML methods capture and "learn" the hidden and discriminative patterns contained by spam/ham mails by utilizing of real world training data. As pointed out by [4], spam filtering is difficult due to its dynamic nature. Thus, the approach of ML has led to create more adaptive and dynamic spam countermeasures in recent decade.

To date, literature of spam mail detection has witnessed the increasing number of machine learning based studies which examine various types of machine learning methods on different types of features. For instance, [5] has employed the method of Naive Bayes whereas [6, 7] have examined the performance of Support Vector Machine on spam email classification. Further, the capabilities of decision trees [8] and conventional neural networks [9] have also been addressed.

In machine learning, it is a well-known fact that, apart from the employed method it also plays a key role that which features have been selected and how they have been preprocessed and represented. If the literature is reviewed, it can be seen that bag of words (BOW) and n-gram feature representations come into prominence since emails are composed of heavily textual data (e.g. words and characters). It is very common in literature to take all the words in the corpus and generate BOW vocabulary for further creation of word frequency based sparse feature vectors that will be used to train a ML model. Besides, character or word level n-gram indexes have also been used frequently. According to Moon et al. [11], n-gram indexes remove the necessity of morpheme analysis and word spacing problems. Moreover, n-gram indexes enable language independency.

In their study, Moon et al. [11] proposed a spam/ham mail classification schema by using n-gram indexing and SVM. They founded out that SVM achieves better results compared to other methods such as Naïve Bayes and *k* nearest neighborhood classification. Blanco et al. [12] have particularly studied the problem of reducing the number of false positives in anti-spam email filters by use of SVM. They proposed an ensemble of SVMs that combines multiple dissimilarities. Dai et al. [13] proposed a system which extracts text based features from body contents and employs an active machine learning method in order to classify malicious spam emails. On the other hand, use of URL (Uniform Resource Locator) has been also incorporated in phishing/spam mail recognition. For instance, Basnet and Doleck [14] have employed URL based features to classify whether a web page is phish or not. They have encoded URLs into 138 dimensional feature vectors and applied several ML methods covering decision trees, random forest, multilayer perceptron and SVM.

As can be seen, various features and different ML methods have been employed in the spam mail classification field. However, according to our best knowledge in spam mail classification field, pure use of hyperlink texts located in email bodies has not been addressed yet. In this study, in contrast to other existing approaches, we propose the use of hyperlink texts located in body of emails as the feature source. In detail, we have employed n-gram indexing schema by extracting word level unigram, bigram and trigram features from a large scale email link corpus collected from 50000 hyperlinks. Extracted n-grams have been filtered out by a certain frequency threshold and then 3 ML methods (Naïve Bayes, SVM and Pegasos - an online version of SVM) have been employed to train 3 different classifiers. According to the results, link texts found suitable to be used as a spam/ham mail classifier. In this way, it is also enabled to have a privacy preserving classification scheme since our approach does not require full body content. We believe that, the proposed approach is a suitable and robust solution for the companies which care about email privacy.

The organization of the paper is as follows: Section 2 briefly introduces the employed ML methods. Section 3 presents the properties of used dataset. Section 4 presents the preprocessing stages and Section 5 details the conducted experiments along with results and finally Section 6 concludes the study.

II. Methods

In this study we have employed three different machine learning methods: (1) Naïve Bayes multinomial (NB), (2) SVM [15] and finally (3) SVM Pegasos [16]. We have first tested the performance of offline methods such as NB and SVM and then moved to Pegasos whether the proposed approach is suitable for an online learning schema. These 3 methods have been briefly explained below. Due to the space limitations, further details about the methods have been left to readers.

A. Naive Bayes (NB)

Naïve Bayes classifiers have been perhaps the most well-known and well-used statistical based classifiers in the field of spam filtering over the years [4]. According to the [5], the main reason of why it is called "Naïve" is that it transforms multivariate problems to a univariate problems by disregarding the possible dependencies or correlations among inputs. As stated in [4], construction and interpretation of NB classifiers are easy and they are effective even for large scale datasets.

Inspired from Bayes' theorem, Bayes classifiers are simple probabilistic classifiers based on strong independence assumptions and they employ a probabilistic approach for inference. Given the class label *y,* a Naïve Bayesian classifier's task is to approximate the class-conditional probability with assumption of conditional independence among the attributes. The conditional independence assumption is given in Eq. 1: [17].

$$P(\boldsymbol{X}|Y=y) = \prod_{i=1}^{d} P(X_i|Y=y) \qquad (1)$$

where each attribute set $\boldsymbol{X} = \{X_1, X_2, \ldots., X_d\}$ comprises d attributes. At this point, instead of calculating class-conditional probability for each combination of $\boldsymbol{X}$, it is sufficient to approximate the conditional probability of each X_i, given Y [17].

For classification of an unknown case, NB classifier calculates the posterior probability of every class Y as presented in Eq. 2 below:

$$P(Y|\boldsymbol{X}) = \frac{P(Y)\prod_{i=1}^{d} P(X_i|Y)}{P(\boldsymbol{X})} \qquad (2)$$

For further reading, readers can see [3] and [17].

B. Support Vector Machine (SVM)

Support Vector Machines (SVM) developed by V. Vapnik is one of the most widely used ML method due to having solid theoretical background along with good generalization performance in several problem domains. Moreover, SVM is also capable of handling high dimensional data avoiding the curse of dimensionality [17]. As a non-parametric method, the principle idea of SVM is building a maximal margin hyper plane between the positive and negative samples in order to linearly separate them. Thus, a better generalization is obtained. A formal explanation about finding a linear separation boundary is given as below [4, 17]:

Let $X = \{(\boldsymbol{x}_i, y_i)\}$, $\boldsymbol{x}_i \in \mathbb{R}^m, y_i \in \{-1, +1\}$ corresponds to the training data which has the collection of attributes where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in})$. Then the separating hyperplane of a linear classifier can be defined as $\mathbf{w}\,.\mathbf{x} + b = 0$ where $\mathbf{w}$ and b constitute the parameters of the model. In this way, a separating hyperplane which divides the training data into corresponding classes is achieved (i.e. $\text{sign}(w^T x_i + b) = y_i$ for all i). At this stage, the margin m_i between a training example and the separating hyperplane can be formulated as in Eq. 3.

$$m_i = \frac{|\mathbf{w}^T\mathbf{x}_i + b|}{\|\mathbf{w}\|} \quad (3)$$

The key point of SVM training phase is to maximize the margin for each example which yields the minimization of the objective function of $f(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2}$ subject to $y_i(\mathbf{w}.\mathbf{x}_i + b) \geq 1$ where i = 1, 2, … *N.* At this point, the standard Lagrange multiplier method can be employed to solve this quadratic and convex optimization problem by use of Eq. 4 presented below:

$$L_p = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \lambda_i \left(y_i(\mathbf{w}.\mathbf{x}_i + b) - 1\right) \quad (4)$$

As a sum, SVM can be expressed as a constrained quadratic programming problem.

As is known, spam has a dynamic nature and a suitable countermeasure should be adaptive for new unseen cases. Though they present satisfactory classification performance the main shortcoming of offline algorithms such as SVM and NB is that they require training from scratch for new examples. In order to overcome this problem, Shalev-Schwartz et al. [16] have extended SVM in terms of (1) better accuracy, (2) shorter training duration and (3) online learning capability and named their own implementation as "Pegasos". Details of Pegasos algorithm can be found at [16]. According to [16], Pegasos is well suited for text classification problems which involve training with sparse feature vectors. Therefore, we have also conducted experiments by employing the SVM Pegasos in order to achieve an easy to update spam email classification schema since it serves an online learning capability which prevents full model training from scratch.

III. Dataset

According to our best knowledge, although there exist various datasets in literature, there exists no publicly available large scale spam/ham email hyperlink dataset. Therefore, the spam/ham email dataset that has been utilized during the study has been supplied by a network security company named COMODO Inc. [18]. The used dataset involves 150000 spam and 141414 ham hyperlinks extracted from 47382 spam and 8506 ham emails collected between August and November of 2016. Supplied dataset comprise 4 attributes: (a) *message-id*, (b) *hyperlink URL*, (c) *hyperlink text* and (d) *spam/ham flag*. As can be seen, dataset does not involve any kind of non-hyperlink information such as subject of message content. In the next stage, whole dataset has been preprocessed in order to generate robust and meaningful n-grams for further analyzes.

IV. Preprocess

Prior to generation of n-gram vocabulary we have employed two phase preprocessing. In the first one, we have checked all instances in spam/ham dataset for the validity of hyperlink texts and found out some inappropriate entries such as null values or HTML/CSS related tags. Therefore, we have cleared those records from the dataset. In this way, the number of spam and ham hyperlinks have been reduced to 146288 and 132439 respectively.

At the second stage of preprocessing we have carried out some sub procedures such as stop word removal (e.g. the, www, and etc.) and uppercasing the words in order to obtain a uniform representation. The procedure that we have followed is given in Table 1 as a pseudo code format.

TABLE I. Pseudo code of the preprocessing procedure

```
Input: D: spam and ham mail dataset(in "message-id, hyperlinkURL, hyperlinkText" format)
Output: PD: Preprocessed dataset
foreach line in D
    select hyperlinkText
    If(line has hyperlinkText)
        add to PD
    else continue;
foreach line in PD
    Uppercase the line
    Divide line to word and add wordlist
    If (word have special characters)
        Remove special characters from word
    If (word length <=2)
        Remove from wordlist
    If (word is a special word) //special word for example; "http, www, and etc.)
        Remove from wordlist
```

Following the preprocessing, we have extracted unigrams, bigrams and trigrams by using a specially created C# application. Since the number and quality of the features have a great impact on machine learning models we have preferred to filter out infrequent n-grams with a certain threshold values such as 30, 40 and 50. Thus, different number of n-gram features have been obtained (See Table 2).

TABLE II. Total number of n-grams with certain thresholds

	Number of n-grams		
Threshold	**30**	**40**	**50**
# of 1-grams	3332	2654	2192
# of 2-grams	4466	3347	2648
# of 3-grams	4474	3267	2540

At this stage, 9 different training datasets belonging to each n-gram and threshold configuration have been generated by use of BOW mapping on detected n-grams. On the other hand, in order to reduce the required training time and hold some remaining data for further validation processes, size of datasets have been reduced to that each contain 50000 instances by using random sampling technique.

V. Experiments and results

As it was stated before, one of the main goals of this study is to understand whether the hyperlink texts in emails play a discriminative role in spam/ham classification. In order to validate this hypothesis we have employed 3 different classification method namely Naïve Bayes, SVM and SVM Pegasos.

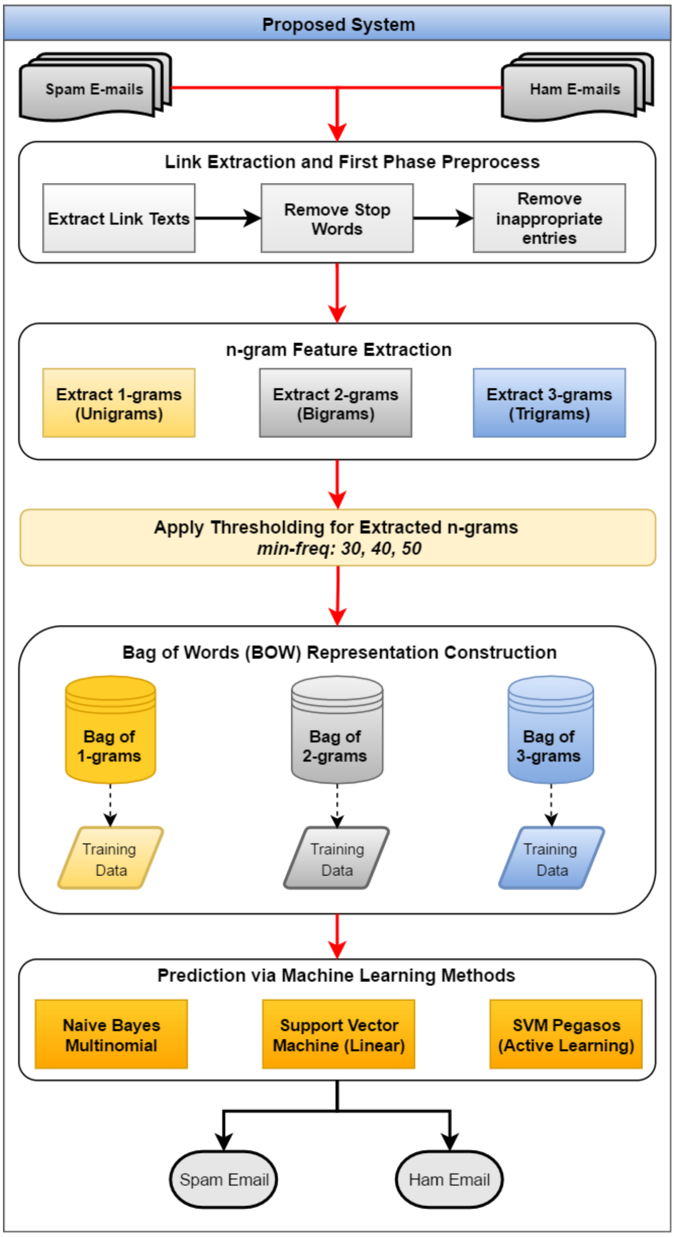
Proposed System
Spam E-mails
Ham E-mails
Link Extraction and First Phase Preprocess
Extract Link Texts
Remove Stop Words
Remove inappropriate entries
n-gram Feature Extraction
Extract 1-grams (Unigrams)
Extract 2-grams (Bigrams)
Extract 3-grams (Trigrams)
Apply Thresholding for Extracted n-grams
min-freq: 30, 40, 50
Bag of Words (BOW) Representation Construction
Bag of 1-grams
Bag of 2-grams
Bag of 3-grams
Training Data
Training Data
Training Data
Prediction via Machine Learning Methods
Naive Bayes Multinomial
Support Vector Machine (Linear)
SVM Pegasos (Active Learning)
Spam Email
Ham Email

Fig. 1. Workflow of the proposed system

Furthermore, SVM and Pegasos algorithms have been adjusted in order to be employed with linear kernel. Throughout the experiments, we have applied 5 fold validation schema to validate stability and robustness of the created models. The workflow of the study has been depicted in Fig. 1. It should be noted that, training of selected ML models have been carried out in software of Rapid Miner Studio 7.5 because of its easy to use GUI based visual modelling tools. Performance of the models have been presented in Table 3 in terms of accuracy and classification error.

TABLE III. AVERAGE ACCURACY SCORES OF 5 FOLD CROSS VALIDATION

Dataset	**# of n-grams**	**Accuracy of Methods**		
		SVM	Pegasos	NB
30 Threshold Dataset	1 Grams	85.92%	90.77%	89.09%
	2 Grams	93.66%	96.02%	95.77%
	3 Grams	89.48%	98.60%	98.45%
40 Threshold Dataset	1 Grams	86.23%	90.76%	88.76%
	2 Grams	94.64%	96.01%	95.74%
	3 Grams	92.37%	98.58%	98.52%
50 Threshold Dataset	1 Grams	86.50%	90.43%	88.58%
	2 Grams	94.43%	96.11%	95.84%
	3 Grams	90.83%	**98.75%**	98.67%

TABLE IV. AVERAGE CLASSIFICATION ERRORS OF MODELS

Dataset	**# of n-grams**	**Accuracy of Methods**		
		SVM	Pegasos	NB
30 Threshold Dataset	1 Grams	14.08%	9.23%	10.91%
	2 Grams	6.34%	3.98%	4.23%
	3 Grams	10.52%	1.40%	1.55%
40 Threshold Dataset	1 Grams	13.67%	9.24%	11.24%
	2 Grams	5.36%	3.99%	4.26%
	3 Grams	7.63%	1.42%	1.48%
50 Threshold Dataset	1 Grams	13.50%	9.57%	11.42%
	2 Grams	5.57%	3.89%	4.16%
	3 Grams	9.17%	**1.25%**	1.33%

According to the obtained results, the following findings have been discovered:

- Among the methods that we have employed, SVM Pegasos has achieved the best result (i.e. accuracy of 98.75%) within the trigram configuration on 50 threshold dataset.
- It has been observed that the classification errors tend to decrease with the increment of threshold value. This finding not only increases the accuracy but also enables building more compact feature vectors.
- Naïve Bayesian classifiers have achieved the least training durations among the employed methods. However, SVM Pegasos seems to be the most feasible method when the capability of online learning is considered.
- Although both SVM and SVM Pegasos have originated from the same computational methodology, it is observed that the improvements in SVM Pegasos enables to collect better results in sparse and linearly separable datasets. Therefore, we can conclude that SVM Pegasos better suits to our problem domain than conventional SVM.
- Apart from the reported findings, we have also repeated the experiments with 10-fold cross validation. Nonetheless, it has been observed that 10 fold cross validation results do not show a significant difference in terms of accuracy.

VI. Conclusion

In this study, it was aimed to use hyperlink texts found in emails in order to classify spam/ham emails. In essence, the usability of the n-gram features extracted from hyperlinks in spam mail classification has been investigated. For this purpose 3 well-used and well-known predictive machine learning methods have been employed. Throughout the study, a novel large scale data set has been utilized. Besides, n-gram indexing schema along with bag of n-grams have been used as the feature generation technique. In order to refine and reduce the large number of features, certain threshold values have been applied.

According to the promising results, hyperlink texts have been found a suitable source of information for spam/ham email classification problem. Moreover, compared with the accuracy values of other studies employing body or header contents, the proposed approach constitutes an effective and feasible alternative for spam email classification. As a future work, it is being planned to use more effective machine learning methods such as Twin SVM or Deep Random Forests in order to achieve more accurate classification.

Acknowledgment

This work is supported by Comodo Group, Inc. and authors would like to thank for their collaboration on dataset collection stage.

References

[1] D. Sculley and G. M. Wachman, “Relaxed Online SVMs for Spam Filtering,” in *SIGIR'07*, 2007.

[2] S. Teli and S. Biradar, “Effective Spam Detection for Email,” in *International Conference on Advances in Engineering & Technology*, 2014.

[3] K. Tretyakov, “Machine learning techniques in spam filtering”, Technical Report, Institute of Computer Science, University of Tartu, 2004.

[4] A. Bhowmick and S.M. Hazarik. Machine Learning for E-mail Spam Filtering: Review, Techniques and Trends, June, 2016. [Online]. Available: https://arxiv.org/abs/1606.01042 [Accessed May 12, 2016].

[5] M. Sahami, S. Dumais, D. Heckerman, E. Horvitz, “A Bayesian Approach to Filtering Junk Email,” in *15th National Conference on Artificial Intelligence*, USA, 1998.

[6] M. Woitaszek and M. Shaaban, “Identifying Junk Electronic Mail in Microsoft Outlook with Support Vector Machine” in *Proceedings of the 2003 symposium on applications and the internet*, 2003.

[7] O. Amayri and N. Bouguila, “A Study of Spam Filtering using Support Vector Machines”, *Artificial Intelligence Review,* vol. 34, pp.73-78, 2010.

[8] F. Toolan and J. Carthy, “Feature Selection for Spam and Phishing Detection,” in *eCrime Researchers Summit*, 2010.

[9] C. Wu, “Behaviour-based Spam Detection using a Hybrid Method of Rule-based Techniques and Neural Networks”, *Expert System with Applications,* vol. 36, pp. 4321-4330, 2009.

[10] L.F. Cranor and B.A. Lamacchia, “Spam!”, *Communications of the ACM*, vol. 41, 1998.

[11] J. Moon, T. Shon, J. Seo, “An Approach for Spam E- Mail Detection with Support Vector Machine and N Gram Indexing,” in *International Symposium on Computer and Information Sciences*, ISCIS, pp. 351-362, 2004.

[12] A. Blanco, A. M. Ricket and M. M. Merino, “Combining SVM Classifiers for Email Anti-spam Filtering,” in *International Work-Conference on Artificial Neural Networks, Computational and Ambient Intelligence,* pp. 903-910, 2007.

[13] Y. Dai, S. Tada, T. Ban, J. Nakazato, J. Shimamura and S. Ozawa, “Detecting Malicious Spam Mails: An Online Machine Learning Approach,” in *International Conference on Neural Information Processing Neural Information Processing*, pp. 365-372, 2014.

[14] R.B. Basnet and T. Doleck, “Towards Developing a Tool to Detect Phishing URLs: A Machine Learning Approach,” in: *2015 International Conference on Computational Intelligence & Communication Technology*, 2015.

[15] C.-C. Chang and C.-J. Lin. “LIBSVM: a library for support vector machines”, *ACM Transactions on Intelligent Systems and Technology*, vol. 2, 2011.

[16] S. Shwartz, Y. Singer, N. Srebro “Pegasos: Primal Estimated sub-Gradient Solver for SVM,” *in 24th International Conference on Machine Learning*, Corvallis, 2007

[17] P. N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Addison-Wesley, USA, 2005.

[18] Comodo Inc, New Jersey, www.comodo.com