A New Efficient KEA Based Approach for E-mail Spam Detection

Qusai Abuein[1] Hassan Najadat[1] Sana Wedian[1,2] Deya' Alzoubi[1,2]
[1]Department of Computer Science
Jordan University of Science and Technology
[1]qabuein, najadat@just.edu.jo
[2]sawedian07@cit.just.edu.jo
[2]deyaazoubi07@cit.just.edu.jo

Abstract *- With the increased advancement in technology and the proliferation of internet applications, electronic mails become an increasingly essential means of communication, for both individuals and organizations.*
In the recent years, however, spam's become the most unsolicited forms of messages that invade the most important services of internet: Electronic mail and search engines. The last decade witnessed a great reaction towards these annoying spam's by applying a variety of filtering techniques to combat spammers, based on the assumption that in any spam mail, there are specific words/patterns that provide indications of spam's, that is, there are predefined barriers that distinguish a spam message from a legitimate one. The most used techniques for spam detection are Bayesian Classifiers that have been argued to be efficient for filtering E-mail spam. However, as spam detection techniques evolve, spammers evolve too so as to prove their excellence to adapt to the predefined barriers. Therefore, content- based recognition techniques must be applied on e-mails to detect spams based on the semantic of their contents. In this paper, we propose a new approach for e-mail spam detection based on k-means algorithm and the Keyphrase Extraction Algorithm (KEA). Our approach achieves high classification accuracy in the sense that it takes into consideration the semantic nature of the textual contents.

Keywords: *E-mail Spam, Keyphrase Extraction, KEA, Semantic Content.*

1 Introduction

As electronic commerce (E-commerce) applications evolve, electronic mails (E-mails) become an enticing target for all different kinds of traders and marketers to advertise their products for sale and their marketing schemes. Moreover, e-mails become an enticing target for spammers to embed their spam content. SpamCon Inc. [1] estimated the cost induced by resources loss and spam filtering associated with only one unsolicited message is 1$ up to 2$ multiplied by the number of spam sent and received every day, the one dollar becomes million.

To solve the problem of spam, there have been several attempts to detect and filter the spam email on the client-side. In previous research [2, 3], many machine learning (ML) approaches are applied to the problem, including Bayesian classifiers as naive Bayes and support vector machine (SVM) [4, 5]. In these approaches, Bayesian classifiers obtained good results by many researchers so that it is widely applied into several filtering softwares. However, most of these filters depend mainly on the pure content of the e-mail message to distinguish between spam and non-spam e-mails. The disadvantage of these methods is that they do not take into consideration the semantic of the contents. Filtering based on contents only is not enough because there are many types of spam e-mail that are well written and designed in such a way to pass the filter without even being classified as spam. For example, advertisements that are used either for the purpose of making money or selling something or for the purpose of spreading hoaxes or rumors. Moreover, there are e-mails that contain web bugs in the form of graphics attached to the e-mail message designed to monitor who is reading the message. Therefore, some of spam e-mails are judged to be non-spam, even if they are truely are. In addition, all types of Content-based spam filters have false positives (these are non-spam messages that are classified as spam); generally, it is more sever to misclassify a legitimate message as spam than to let a spam message pass the filter [2].

Another serious problem is that what is classified as spam by these filters may not truly be so because spam is a relative concept, that is; what might be considered as a spam for one person may not be so for another one. Further issue which most of such approaches ignore is the semantic meanings of the textual contents; i.e. in any document, there is one or more phrases whose existence gives an indication of the topic or the general idea of that document.

In this work, we will apply our new approach that takes the semantic of the content into consideration on e-mail spam detection. Our approach is a combination of the well-known clustering algorithm (k-means) and the Keyphrase Extraction Algorithm (KEA) that is mainly used for extracting the most important keyphrase from a specific document and for documents summarization [6].

2 Related Works

Due to the serious problems associated with spam, a number of automated filtering approaches were proposed in the literature to overcome such problems. Early proposed approaches for spam filtering relied mainly on manually constructed pattern-matching rules that need to be tuned according to each user's message [7]. These approaches allow users to hand-build a dataset that is based on a set of logical rules to detect spam e-mails. However, these approaches are seemed to be tedious and problematic, since users need to pay a full attention just to build the desired set of rules. In addition, it is a time consuming process, since the generated set of rules should be changed or refined periodically as the nature of spam changes too.

To overcome the problems associated with the manual construction of rules, another approach was proposed by Cohen et al [8] so as to automatically adapt to the changing nature of spam over time and to provide a system that can learn directly from data already stored in the web server databases. These approaches are efficient when applied for general classification tasks, that is; the classification of e-mails is either spam or non-spam based on their text, with no regards to the existence of some domain specific features.

Sahami et al [2] were one of the pioneers who applied Naïve Bayes algorithm as an example on machine learning algorithms for the purpose of spam filtering. In this algorithm, the Naïve Bayes classifier learns to classify documents into fixed classes (spam and no spam), based on their content, after being trained on manually classified documents. It reported surprisingly good results in terms of precision and recall; however, it had a high ratio of false positives (i.e. non-spam messages classified as spam).

Minoru et al [9] have proposed an e-mail spam detection using text clustering based on vector space mode. In this scheme, the contents of various kinds of e-mails are represented as several term statistics as the centroid vectors rather than one term statistics in Naive Bayes Classifier. The basic idea is to represent e-mails as vectors using vector space model. Then the vectors of e-mails are divided into clusters using k-means algorithm, such that a cluster mean (centroid vector) is obtained to act as a representative for each cluster (this will facilitate the classification later), then a label spam/nonspam is assigned to each cluster centroid vector by calculating the number of spam e-mails in a cluster. If the number reaches a threshold the a spam label is assign to it, otherwise; nonspam label is assigned. After that, whenever a new e-mail needed to be classified, it is first represented as vector, then it's similarity to the centroid of each cluster is computed (using one of the similarity measures as cosine similarity). The e-mail will be classified into the cluster whose centroid is much closer than the other one, and a label of the most relevant cluster is assigned to the new e-mail.

Other methods depend on statistical techniques to detect spam e-mails [10].

3 K-means Algorithm

K-means [11] is one of the simplest unsupervised learning algorithm to cluster *n* objects into *K* partitions based on specific attributes, where $K<n$, in which the number of clusters to be created is fixed a priori by the user. The algorithm proceeds by randomly defining *K* centroids and assigning a document to the cluster that has the nearest centroid to the document.

The following steps describe the basic K-means algorithm to find *K* clusters:

Step 1: Select *K points* as the initial centroids for each cluster.
Step 2: Every point of the data set is assigned to its nearest centroid.
Step 3: Re-calculate the centroid of each cluster.
Step 4: Repeat Steps 2 and 3 until the centroids no longer move.

4 Keyphrase Extraction Algorithm (KEA)

Keyphrases are the important words/phrases that reflect the subject of the text documents and provide semantic metadata that summarize and characterize documents. Keyphrases are particularly useful because they can be interpreted individually and independently from each other. They can be used in information retrieval systems as descriptions of the documents returned by a query, and as a document clustering technique.

Keyphrase extraction is a method of automatically extracting important phrases from a text mainly for document summarization. KEA [12] is a Java-based keyphrase extraction algorithm that generates candidate phrases from a document and selects keyphrases from them by using TF-IDF weight and naive Bayes classifier.

KEA's extraction algorithm has two phases, training and extraction. The training phase uses a set of training documents for which the author's keyphrases are known. For each training document, candidate phrases are identified and their feature values are calculated. Each phrase is then marked as a keyphrase or a non-keyphrase, using the actual keyphrases for that document. In the extraction phase, the algorithm chooses keyphrases from a new document using the above model. To select keyphrases from a new document, KEA determines candidate phrases and feature values, and then applies the model built during training. The model determines the overall probability that each candidate is a keyphrase, and then a post-processing operation selects the best set of keyphrases. Both phases choose a set of candidate phrases from their input documents, and calculate the values of certain attributes for each candidate.

This process of Kea is shown in Figure 1 [2]. Both stages choose a set of candidate phrases from their input documents, and calculate the values of certain attributes for each candidate.

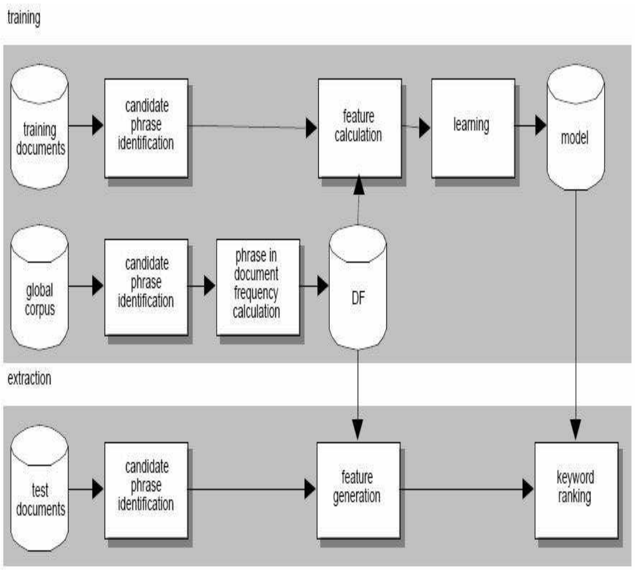
training
training documents
candidate phrase identification
feature calculation
learning
model
global corpus
candidate phrase identification
phrase in document frequency calculation
DF
extraction
test documents
candidate phrase identification
feature generation
keyword ranking

Figure 1: The Key Phrase Extraction Algorithm (KEA)

5 *Proposed Algorithm*

In this section, we propose our approach for spam e-mail detection and filtering. Our proposed algorithm is a hybrid of two well known algorithms, the K-means and the Keyphrase Extraction Algorithm (KEA). To obtain the spam detection, our algorithm performs dual task, keyphrase extraction (based on KEA) and clustering (based on k-means).

In the context of spam filtering, our approach can perform efficiently and provides promising results. Considering the task of spam filtering as a clustering problem, we can apply this algorithm on a trained dataset (i.e. general corpus of e-mail spam) to obtain (extract) a list of keyphrases or keywords that uniquely identify the message and indicate its spaminess (i.e. whether it is a spam or non-spam) and use such list to build a model that can be trained further for filtering purposes.

The idea is to apply KEA algorithm into the documents collection to extract the most indicative keyphrases, then the feature values for each keyphrase are calculated using **TFxIDF** where Term Frequent (**TF**) is the number of times keyphrase (*i*) appears in a documents *Dj* and Inverse Document Frequency **IDF = Log (N/ni)** where *N* is number of documents in a collection and *ni* is number of documents that contain *ith* keyphrase, then the whole document is given a weight based on the result. Then we apply K-means algorithm into a set of weighted documents to create clusters with corresponding centroid for each cluster, calculate the distance-using the cosine similarity measure between each document and the centroid, and assign the document to the cluster whose centroid is the nearest to that document. After that the operations of k-means algorithm are performed repeatedly until no further change happens on a cluster. When new e-mail is received it is applied on KEA to extract keyphrase then compute the weights for each keyphrase and assign weights to document vector, calculate the similarity between the document vector and the nonspam and spam centroid then assign e-mail to nearest centroid. The steps of our proposed approach are given as follows and are illustrated in Figure 2:

1. Apply KEA algorithm to each document (e-mail) in the data set to extract key phrases from the selected document.
2. Represent e-mail as a document vector with values (x_1, x_2 ….. x_m) of attributes (X_1, X_2.... X_m), where Xi are the extracted key phrases and i=1,2,3 …, m.
3. Compute the feature values for each keyphrase and assign weights to document vector
4. Set the number of clusters K to 2
5. Apply the K-means algorithm to generate K clusters with K centroids.
6. Calculate the similarity between the documents and the centroid.
7. Assign the documents to the cluster that has the nearest centroid.

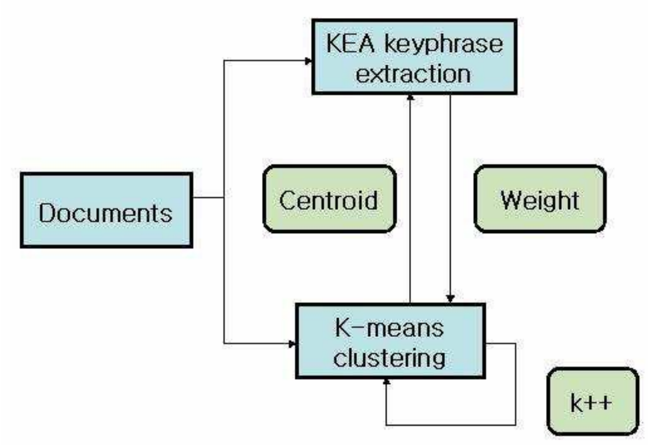

Figure 2: The Proposed KEA-based Algorithm

6 *Experiments and Results*

To evaluate our system efficiently for detection spam e-mails we use the freely available Ling-Spam collection [9]. The Ling-Spam collection consists of 2412 nonspam messages and 481 spam messages by hand categorization.

6.1 *Experimental Environment*

By using stop-list and lemmatizer, this collection consists of:

- Bare (untreated)
- Lemm (using lemmatizer)

- Stop (using stop-list)
- Lemm + stop (using lemmatizer + stop-list).

We used bare for our experiments. Then we employed accuracy measures to evaluate the performance of our proposed approach, these measures are precision and recall, which are defined as follows:

$$\text{Recall} = \frac{truePositive}{truePositive + falsePositive} \qquad \textbf{(1)}$$

$$\text{Precision} = \frac{trueNegative}{trueNegative + falseNegative} \qquad \textbf{(2)}$$

Accuracy =

$$\frac{truePositive + trueNegative}{truePositive + falsePositive + trueNegative + falseNegative} \qquad (3)$$

Where:

- **True positive**: number of spam Emails that are correctly classified.
- **True negative**: number of nonspam Emails that are correctly classified.
- **False positive**: the number of nonspam Emails classified as spam.
- **False negative**: number of spam Emails classified as nonspam.

6.2 Results

The result show that our approach achieves high accuracy comparing to approach of [8] in the sense that it takes into consideration the semantic nature of the textual contents, although we only need 2 clusters according to our approach , while the approach of [8] had preformed using 50 and more cluster.

In the experiments we applied our approach in 20, 40, 60, and 80 and 100 documents the results show that our algorithm performs good result for detection email spam as shown in Figure 1.

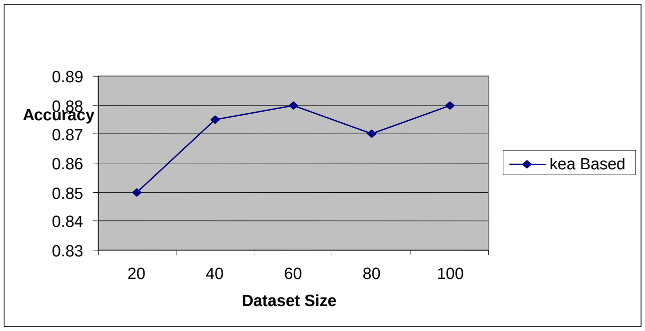

Figure 3: Accuracy versus dataset size

1 Conclusions and Future Works

In this paper we introduced a new approach for e-mail spam detection which is a combination of both k-means algorithm and the Keyphrase Extraction Algorithm (KEA). The results show that our algorithm achieves good result for detection e-mail spam. Further experiments and investigation about using almost the same number of clusters for all approaches (similar environments) are needed.

References

[1] Juhyun Han, Taehwan Kim, Joongmin Choi **,"** Web Document Clustering By Using Automatic Keyphrase Extraction", 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent technology Workshops. Pages: 56-59, 5-12 Nov. 2007.

[2] Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin and Craig G. Nevill-Manning ,"KEA: Practical Automatic Keyphrase Extraction". In Proceeding of the 4th ACM Conference on Digital Libraries, 254-255, August 1999.

[3] L.Zhang, J.Zhu and T.Yao "An Evaluation of Statistical Spam Filtering Techniques" ACM Transactions on Asian Language Information Processing, Vol.3, No.4, December 2004, pages243-269.

[4] S.Atkins "Size and Cost of the problem". In Proceeding.of the Fifty-sixth internet Engineering Task Force(IETF) Meeting, (San Francisco, CA), March 16-21 2003.SpamCon Foundation.

[5] K.Ahmad "An Overview of Content-Based Spam Filtering Techniques", Djillali Liabes University, Journal of Informatica (Slovenia) , Vol. 31 , No. 3 (2007) , p. 269-277.

[6] Cohen, W.W "Learning Rules that Classify E-mail", In Proceedings of AAAI Spring Symposium on Machine Learning in Information Access, Stanford, California, 1996.

[7] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk e-mail. In Proceedings of the AAAI-98 Workshop on Learning for Text categorization, pages 1048–1054, 1998.

[8] Minoru Sasaki, Hiroyuki Shinnou, "Spam Detection Using Text Clustering".In Proceedings of the 2005 International Conference on Cyberworlds (CW'05).Pages 315-319. Nov.2005

[9] ling-spam dataset. http://www.iit.demokritos.gr/skel/i-config/downloads/. Access date (November, 2008*).*

[10] J. B. MacQueen (1967): "Some Methods for classification and Analysis of Multivariate Observations. In Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability", Berkeley, University of California Press, 1:281-297.

[11] I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, and C. Spyropoulos. « An evaluation of naive bayesian anti-spam filtering ». In Proceedings of the Workshop on Machine Learning in the New Information Age: 11th European Conference on Machine Learning (ECML 2000), pages 9–17, 2000.

[12] H. Druker « Support vector machines for spam categorization » . In Proceedings of the IEEE Transaction on Neural Networks, volume 10, pages 1048–1054, 1999.

[13] A. Kolcz and J. Alspector « Svm-based filtering of e-mail spam with content-specific misclassification costs ». In Proceedings of the TextDM!G01 Workshop on Text Mining, IEEE International Conference on Data Mining, pages 1048– 1054, 2001.