

Web Spam Detection Using Machine Learning in Specific Domain Features

Hassan Najadat¹, Ismail Hmeidi²
Department of Computer Information Systems
Faculty of Computer and Information Technology
Jordan University of Science and Technology
Irbid 22110, Jordan
najadat@just.edu.jo¹
hmeidi@just.edu.jo²

Abstract: In the last few years, as Internet usage becomes the main artery of the life's daily activities, the problem of spam becomes very serious for internet community. Spam pages form a real threat for all types of users. This threat proved to evolve continuously without any clue to abate. Different forms of spam witnessed a dramatic increase in both size and negative impact. A large amount of E-mails and web pages are considered spam either in Simple Mail Transfer Protocol (SMTP) or search engines. Many technical methods were proposed to approach the problem of spam. In E-mails spam detection, Bayesian Filters are widely and successfully applied for the sake of detecting and eliminating spam.

The assumption that each term in the document contributes to the filtering task equally to other terms and the avoidance of user's feed back are major shortcomings that we attempt to overcome in this work. We propose an improved Naïve Bayes Classifier that gives weight to the information fed by users and takes into consideration the existence of some domain specific features. Our results show that the improved Naïve Bayes classifier outperforms the traditional one in terms of reducing the false positives and the false negatives and increasing the overall accuracy.

Keywords: Web Spam, Naïve Bayes, Term Frequency Matrix (TFM), Confusion Matrix (CM).

1. Introduction

With the increased advancements in internet applications and the proliferation of information available for the public, the need for efficient search engines that are able to retrieve the most relevant documents that satisfy users' needs becomes evident.

From Information Retrieval (IR) perspective, search engines are responsible for retrieving a set of documents that are ranked in descending order according to their relevancy [2].

A common problem encountered in this context is that there are some documents marked with a high rank and retrieved as the first (or one of the top) documents by the search engines where they are truly not [5]. Several reasons exist to justify this problem; one reason is related to the extent to which a user knows exactly what he or she is searching for, and consequently, his or her knowledge is reflected on the retrieved results.

Another important reason is the existence of the so called: Spam Web pages; these are pages that from the search

engines' point of view seem to be relevant, but in reality they contain no useful information for users [5].

In their discussion about web spam, Castillo et. al. [4] defined web spam as any attempt to deceive a search engine's relevancy algorithm, or an action performed with the purpose of influencing the ranking of the page. Detecting Web Spam is considered as one of the most challenging issues facing search engines and web users [11]. Since the search engines are the gates to the World Wide Webs, it is important to provide the possible best results answering the user's queries. There are some people well known as spammers try to mislead the search engines by boosting their web pages rank, as a result capture user attention to their pages. These pages contain a few or not any useful information that the user expects to find. The search engines need to detect or filter spam pages to provide high quality results to users (i.e. truly relevant pages). For a search engine to be evaluated as an efficient one, it should not only return as much documents as possible, but also should return those relevant documents that are spam-free.

Currently, many techniques are applied by search engines to fight spam, such as detecting spam web pages through content analysis [11]. This technique is the most popular technique for spam detection currently used by search engines such as Google; nevertheless, it is still lack to find all spam web pages. A separate section is devoted to detail this technique further.

Spam can be very annoying in the context of search engine for several reasons. First, in the case there are financial advantages from search engine, the existence of spam pages may lower the chance for legitimate (legal) web pages to get the revenue that they might earn in the absence of spam. Second the search engine may return irrelevant results that users do not expect, and therefore, a non-trivial portion of time might spent on-line wading through such unwanted pages. Finally the search engine may waste important resources on spam pages, this include wasting network bandwidth (Crawling), wasting CPU cycles (Processing), and wasting storage space (Indexing) [11].

Microsoft Researchers [11] show that some particular top-level domains are more likely to contain spam than others do, for example, .biz (Business) has greatest percentage of spam with 70% of all pages being spam, .us domain comes

in second place with 35% spam pages. Moreover, pages written in some particular languages are more likely to be spam than those written in other languages, for instance pages written in French are the most likely to be spam, with the percentage of 25% of being spam.

Spammers proved their excellence to adapt to the different formats available for Web pages, several spamming techniques used by spammers to influence the ranking page algorithms of search engines. All these techniques are considered challenging for web page spam detection algorithm, especially for Contents-Based approach. The two main categories of spammer techniques are: Term Spamming, and Link Spamming [11, 3]. In term spamming, many techniques that modify the content of the page are applied. The content includes: the document body, the title, Meta tags in HTML header, anchor texts associated with URLs and page URLs. The spammers can attach their unsolicited content (i.e. spam) to one or more of these contents resulting in a new page that can pass the spam filter without any doubt of being legal. Among all term spamming techniques, the most popular one is body spamming [11], in which terms are included in the document body, an example is to include specific terms as "Free grant money", "free installation", "Promise you ...!", "free preview", etc.

Another way of grouping term spamming techniques is based on the type of terms that are added to the text fields, either by repeating one or a few specific terms, including a large number of unrelated terms, or stitching phrase wherein, sentences or phrases, possibly from different sources are glued together [11].

In link spamming technique, spammers tend to insert links between pages that are present for reasons other than merit [13]. Link spam takes advantage of link-based ranking algorithms, such as Google's Page Rank algorithm, which gives a higher ranking to a website that is cited by other high ranked websites.

In correspondence to the aforementioned spamming techniques, many content-based spam filtering techniques were proposed. The importance of analyzing the content of a particular web page is that spammers tend to boost their web pages rank by applying spamming techniques on these contents [11]. During content analysis, the number of words in the page's body and title, the average length of words, the amount of text anchor and keywords in metatags are analyzed to detect the abnormalities in these contents that are interpreted as spamming attempts.

The rest of this paper is organized as follows. Section 2 gives an overview of the works related to spam detection. The Naïve Bayes classifier is illustrated in Section 3. Section 4 proposed our approach, and our experimental results are shown in Section 5. Section 6 concludes the paper and provides future directions.

2. Related Work

Due to the important role that web pages occupy as means for supporting electronic commerce (E-Commerce), web

pages become enticing target for all different kinds of traders and marketers to advertise their products for sale, get-rich-on-the-fly schemes [13, 17], and to get information about pornographic web sites. Moreover, they 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.

Because of the serious problems associated with the unsolicited spam contents of either a single E-mail or a large web page, a number of automated filtering approaches were proposed in the literature to overcome such problems [16, 10]. These filters are used mainly for E-mail spam and then transformed to be used in the context of Web Spam.

Early proposed approaches for spam filtering relied mostly on manually constructed pattern-matching rules that need to be tuned to each user's message [9]. That is, they allow users to hand-build a rule set that consists of a set of logical rules to detect spam emails and Web pages. 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, which by the way not all users can build such a set. 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.

Because of the problems associated with the manual construction of rules, another approach was proposed in [7] 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 proved as successful when applied for general classification tasks, that is, the classification of E-mail to either spam or non-spam based on their text, with no regards to the existence of some domain specific features.

Several machine learning algorithms have been proposed for text categorization (classification) [14, 15]. These approaches were investigated to be used for spam filtering since it is viewed as a text categorization problem. In [13], they applied a machine learning algorithm for the purpose of spam filtering. In this algorithm, the filter learns to classify documents into fixed classes (i.e. spam and nonspam), based on their content, after being trained on manually classified documents.

As a variation of the rule-based approaches discussed above, a great deal of work was witnessed in the literature to automatically perform content-based classification. Naïve Bayes classifiers [16], was proposed as a good example of those approaches that showed satisfactory results in the context of E-mail spam Filtering. [13] trained a Naïve Bayes classifier on manually classified spam and nonspam messages reporting surprisingly good results in terms of precision and recall. Our work utilizes Naïve Bayes classifiers based on the context of web pages to detect the spam pages automatically.

3. The Naïve Bayes Classifier

A Naïve Bayesian classifier is a simple probabilistic

classifier based on applying Bayes' theorem with a strong (naive) independence assumption that all variables A_1, A_2, \dots, A_n in a given category C are conditionally independent with each others given C [16]. Depending on the precise nature of the probability model, Naive Bayes classifiers can be trained very efficiently in a supervised learning setting [6,12].

Beside Naïve Bayes classifiers, a variety of supervised machine learning algorithms such as Support Vector Machine (SVM) and memory-based learning [4, 8] have been successfully applied and showed satisfactory results in the context of spam filtering.

Although these techniques discussed above proved to perform well in some cases, they still have problems in other cases: for instance, all types of Content-based spam filters have false positives; generally, it is more sever to misclassify a legitimate message as spam than to let a spam message pass the filter [4]. In addition 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.

These limitations are driving factors for us to develop a novel technique for spam filtering by using a user-oriented feedback mechanism that works in combination with Naïve Bayes classifier to reduce the false positives and false negative encountered by traditional classifiers, in addition, a special concern is given to some specific domain features (terms and patterns) that maybe considered as spam discriminators.

The classifiers can be applied on spam filtering being viewed as text classification problem as follows: Given a set of training documents, $D = \{t_1, t_2, \dots, t_n\}$ of tuples and a set of classes $C = \{C_1, C_2, \dots, C_n\}$. The classification problem is to define a mapping $f: D \rightarrow C$ where each t_i is assigned to one class with their associated class labels, each document, t_i , is represented by a vector of words $\{w_1, w_2, \dots, w_n\}$. The independent probability of w_i of a given document associated with class C can be written as in [6] $p(w_i | C)$.

Since each document consists of a large number of words, the Naïve Bayes classifier makes the simplifying assumption that w_1, w_2, \dots, w_i are conditionally independent given the category C ,

$$P(D | C) = \prod_i p(W_i | C) \quad (1)$$

The probability of a given document D belongs to a given class C is represented as $P(C|D)$, which can be computed

$$P(C | D) = \frac{P(C)}{P(D)} P(D | C) \quad (2)$$

To estimate the probability of a particular document is spam, given that it contains certain words, Bayes' theorem states that the probability of finding those certain words in spam documents, times the probability that any document is

spam, divided by the probability of finding those words in any document [13]:

$$P(spam | words) = \frac{p(words | spam)}{P(words)} p(spam) \quad (3)$$

4. Web Spam Detection Classifier

Finding a spam web page is viewed as supervised text classification problem. In the supervised classification application, the web spam classifier needs to be trained with a set of web pages that are previously classified into two categories, spam and non-spam.

Since spam is a relative concept, that is, what is considered spam for one user may not be the same for other users. Moreover, what might be spam for a specific user at a particular time might not be so for the same user at different time, then, depending only on the capabilities of the trained classifier as the case of many traditional Naïve Bayes Classifiers seems to be of limited benefits.

In training phase, a user-oriented preparation is performed, wherein, the web pages reside in the web server are classified into spam or nonspam based on user's feedback and the automated classification by the filter. The attention is given for the general spam that the majority of users agree upon, then all the terms contained in the pages classified as general spam are extracted to form the General Spam Dictionary, this phase is illustrated in Figure 1.

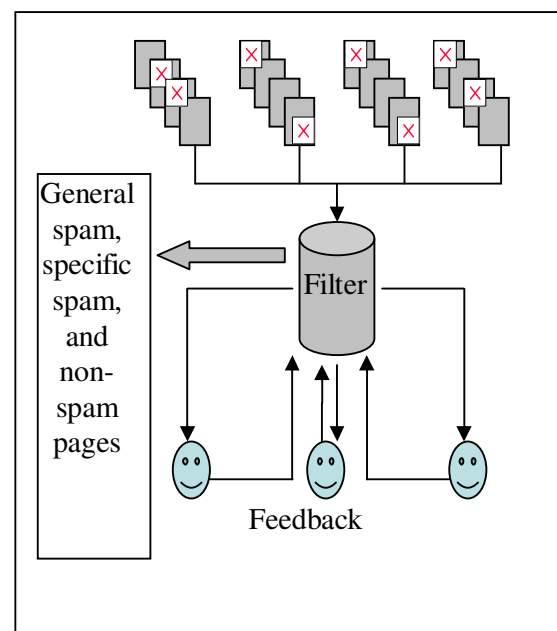


Figure 1. User Oriented Training

Therefore, a novel user-oriented training mechanism is needed to support the classifier with periodic users' feedback to determine whether the page is considered as spam or nonspam. In this training scenario, there are two outcomes: those web pages that are judged to be spam by the majority of users, we call such pages: the General Spam, and those

web pages that viewed to be spam by one or few users, we call them Specific Spam. Our attention will be focused on the general spam, because they can contribute efficiently in the process of classification.

This novel training mechanism is proposed mainly to function in the server-side rather than in the client-side. It is an effective and promising mechanism to overcome the previously mentioned problems associated with web server being affected with spam, especially the problem of wasting resources on spam pages (the resources include server's bandwidth, CPU cycles and storage space).

The spam detection system consists of three phases which include training phase, preprocessing phase, and classification phase.

As shown in Figure 2, the preprocessing phase is the cleaning process, which is applied to each web document to extract its body. In stemming and stop words removal, the frequent words that do not contribute efficiently in classification process are removed from the body of the page.

Stemming operation reduces distinct words to their common stem, which is achieved by removing prefixes and suffixes from words. We choose the affix stemmers algorithm in the stemming work. This assists in reducing the time required for classification since the words length is lessened, which yields to reduce the accuracy of classification.

The vital step in this phase is the generating dictionaries, wherein two different lists of words are generated, such lists are called dictionaries, and they include: the frequent terms dictionary, extracted from those web pages that are classified as non-spam, and the special features dictionary. Each entry in the frequent terms dictionary consists of <term, probability of being spam>.

In classification phase, the preprocessed documents are represented by Term Frequency Matrix (TFM) structure [5] to perform the statistical analysis (i.e. Bayesian rule). TFM simplifies the calculation of the probability of word, belongs to class, $P(\text{class}_j | \text{word}_i)$, and also improves the efficiency of Naïve Bayes classifier which requires only one scan through the entire training dataset. As provided in Figure 3, each intersects of word and class row in Term Frequency Matrix represents the number of times (or frequency) of the word appear in class j .

Assuming that each feature contributes in the process of spam filtering in an equal manner to each other feature may

lead to inadequate results. In other words, there is no particular feature in the text of the web page that provides evidence as to whether the page is spam or nonspam. However, this assumption does not hold in many real situations. For example, it is proved by experience (and from users' feedback) that there are many specific features whose existence provides a strong indication on the suspicious message (spam), such as "free money", "congratulation you are winner number...", "\$\$\$\$ you win xxx\$\$\$\$", and the over used punctuations "?????".

In addition to these discriminating textual features (patterns), web pages contain many non-textual features that indicate whether it is spam or not such as the domain type .edu, .org, .com, .biz, etc [4]. It is shown by Microsoft researchers [11] that 70% of those pages with the domain .biz are spam, and that .edu pages are rarely (or never) contain spam.

To this end, we consider the employment of such (textual and non-textual) features as good discriminators of spam that insist in a correct classification. To achieve this, we maintain a table (or named as dictionary) called specific feature dictionary consisting of all these specific features, each entry in this table corresponds to: <feature, probability>. Then, it becomes straightforward to incorporate such additional features to our Naïve Bayes Model.

As a new web page needs to be classified, a list of all its words is generated and checked against the pre-established features dictionary to make sure whether it contains one or more of its words that are determined to be spam discriminators. In addition to the comparison with the specific features dictionary, the new webpage is compared against other dictionaries (i.e. the general spam and the frequent terms dictionaries). After each comparison, the probability of spam is computed, as a result, we come up with a probability value that takes into account the domain specific features existence, the words that are judged (by users) to be spam, and the ordinary terms that might be spam in some cases (according to their probability).

Figure 4 depicts the Naïve Bayes classifier with user feedback and domain specific features.

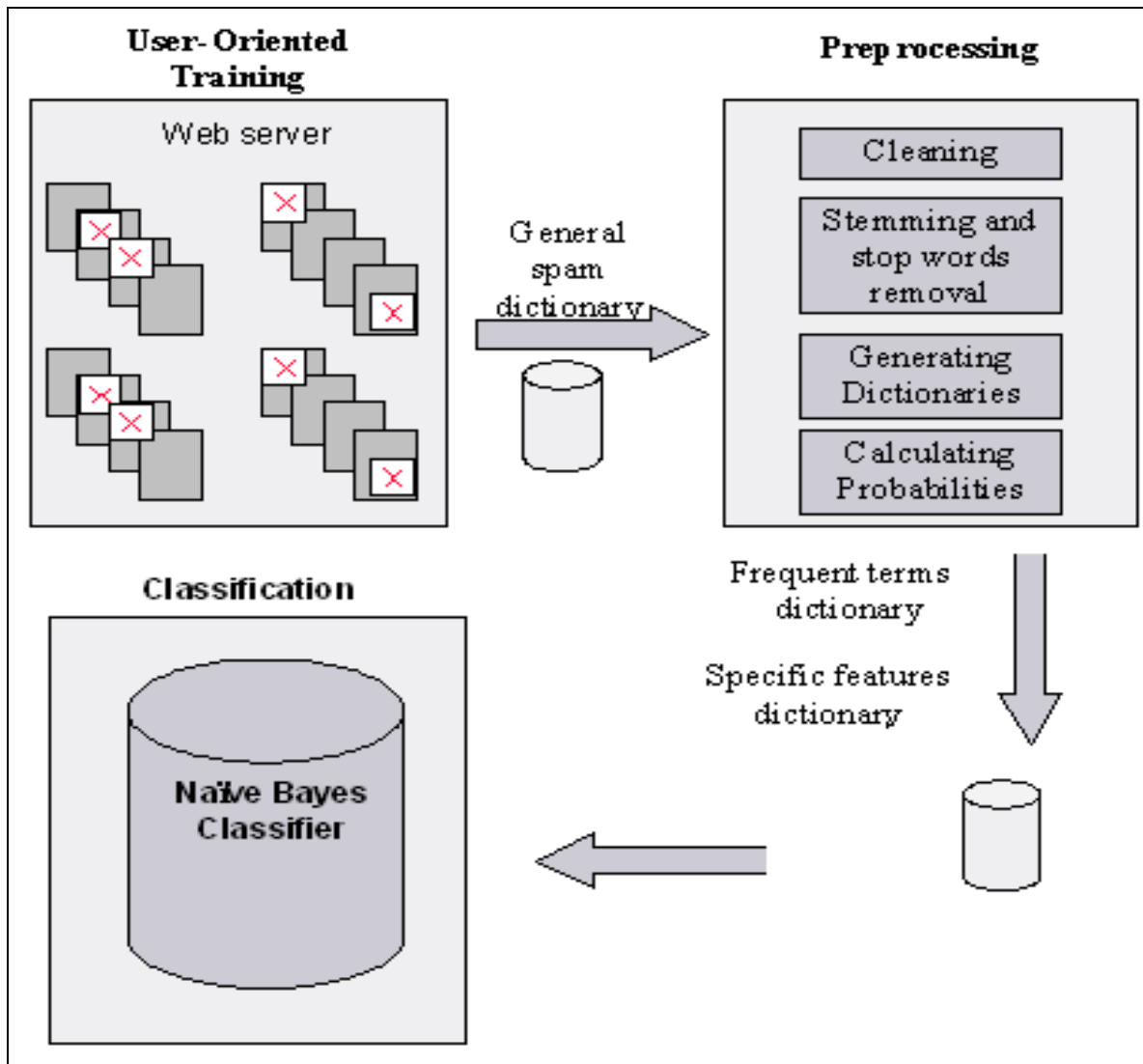


Figure 2. Web Spam detection system structure with Naïve Bayes Classifier

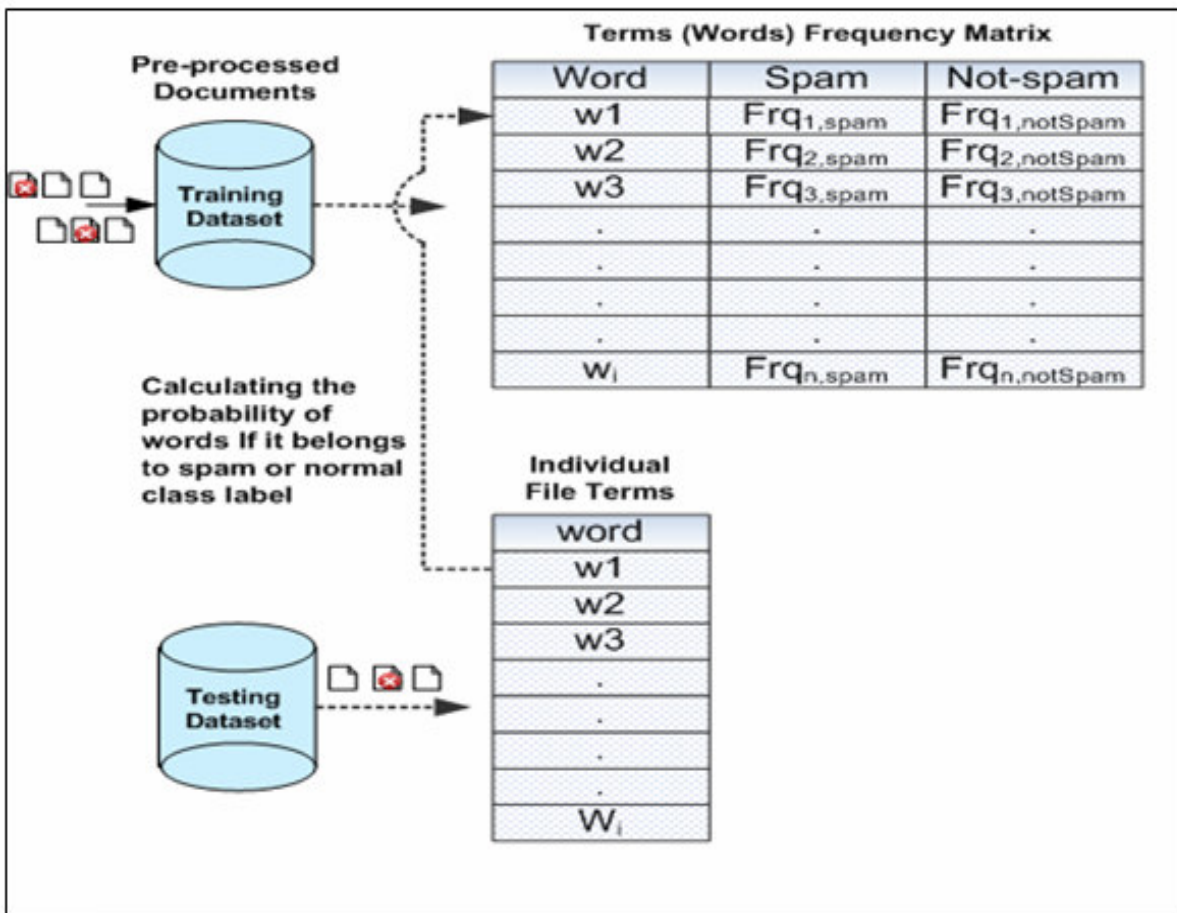


Figure 3. Term Frequency Matrix associated with individual file terms

5. Experimental Results

Our experiments were all performed on the webspam-UK2006 data set [4]. The training dataset consists of 8,1415 web pages. A detailed description of their data set and the criteria in assigning a web to be spam or non-spam can be found in [4]. In our work, a sample of these web pages is taken to evaluate the classification accuracy of our spam detector.

To estimate the accuracy of our proposed algorithm, we use a popular accuracy measure in the context of Information Retrieval, namely: the Confusion Matrix (CM). CM contains information about actual and

predicted classifications done by a classification system [6]. Figure 5 shows confusion matrix for the two classes spam and non-spam. As in [6],

TP represents actually positive and classify as positive,
FN represents actually positive and classify as negative,

FP represents actually negative and classify as positive,
TN represents actually negative and classify as negative.

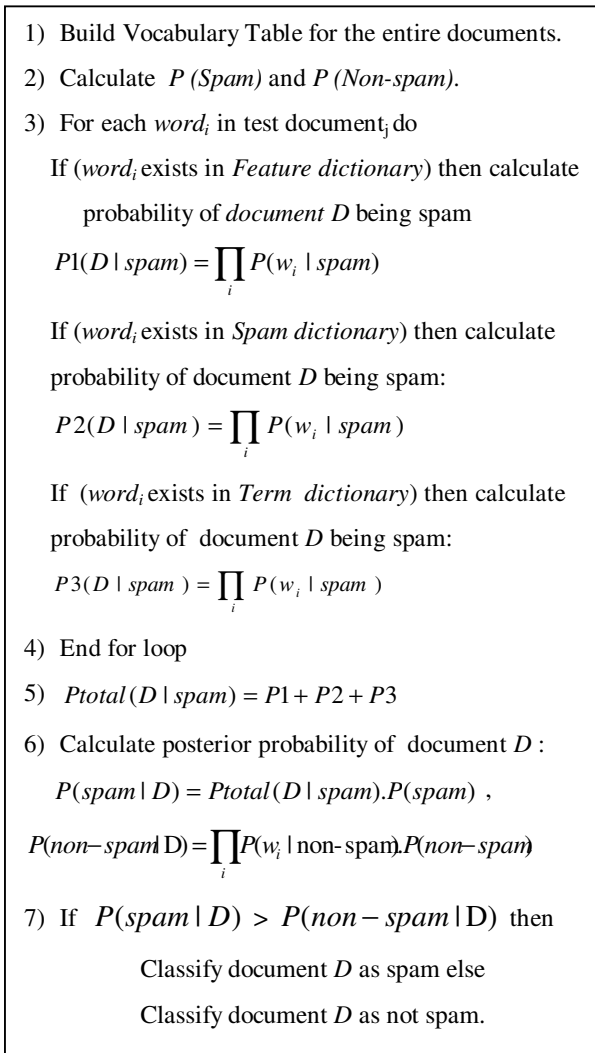


Figure 4. Spam Detection Procedure

		Spam	Nonspam
Actual Class	Spam	TP	FN
	Nonspam	FP	TN

Figure 5. Confusion Matrix

We use the confusion matrix to calculate *sensitivity* and *Specificity* measures. Sensitivity refers to true positive ratio, that is, the proportion of positive documents that are correctly identified. Specificity is the true negative ratio, that is, the proportion of negative documents that are correctly identified. Where truePositive is the number of true positive documents are correctly classify. TrueNegative is the number of true negative documents that are correctly classified. FalsePositive is the number of false positive documents that are incorrectly classified [6].

$$Sensitivity = \frac{truePositive}{truePositive + falseNegative} \quad (4)$$

$$Specificity = \frac{trueNegative}{trueNegative + falsePositive} \quad (5)$$

We use the above measurements to compute the accuracy which is defined as follows:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (6)$$

To evaluate the accuracy, holdout technique is utilized to produce a true estimate of the classifier. The data are partitioned into two separated dataset, training set and testing set. The training set used to learn the classifier algorithm, and the testing set in used to evaluate the accuracy. We run our classifier on different five samples and then calculate the classification accuracy for each run. For instance, we calculate the accuracy of a sample consisting of 238 testing documents (with 69 of documents belong to Nonspam class, and 169 of documents belong to Spam class). For this test, the Sensitivity and Specificity are 97% and 66% respectively. And the total accuracy is 88%, where total accuracy equal $((TP + TN) / (TP+FP+TN+FN))$. Other experiments are made on different samples consisting of 324, 400, 519 and 618 documents.

The accuracy results are shown in figures 6 and 7, these figures show also the effect of stemming the document before classification on the accuracy results. Figure 7 indicates that using stemming documents gains 80% in average, while Figure 6 shows the accuracy is 78%.

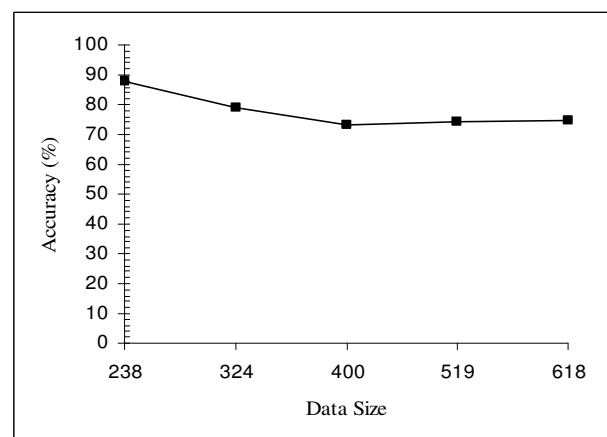


Figure 6. Accuracy versus Dataset size (Stemming text)

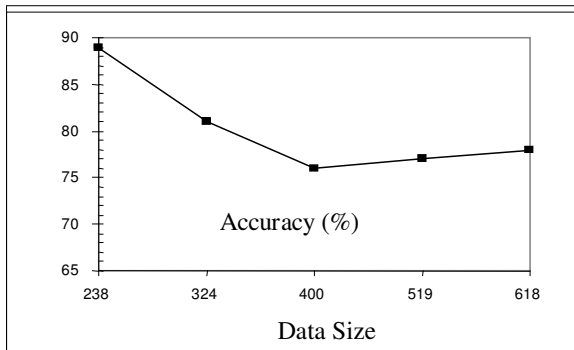


Figure 7. Accuracy versus Dataset size (Unstemming text)

We also evaluate the classifier performance by concentrating on the number of nonspam pages that are wrongly classified as spam, and the number of spam pages that are wrongly classified as nonspam. The first parameter is called the Nonspam Misclassification Rate (HMR) while the second parameter is called Spam Misclassification Rate (SMR).

In the context of spam filtering, it is proved that the effect of misclassifying a legitimate message as spam is more severe than misclassifying a spam message as legitimate. The accuracy is compared for both Naïve Bayes Classifier with Domain Specific Features (NBCDSF) and the Naïve Bayes Classifier without User Feedback (NBCUF), the results show that the improved Naïve Bayes classifier with user feedback outperforms the NBCUF in terms of increased accuracy and decreased SMR and HMR rates. The results are shown in Figure 8.

6. Conclusion and Future work

Web spam pages are an annoying problem that were prevented by many techniques, among them, is the Naïve Bayes classifiers that proved to be efficient mechanisms for spam filtering.

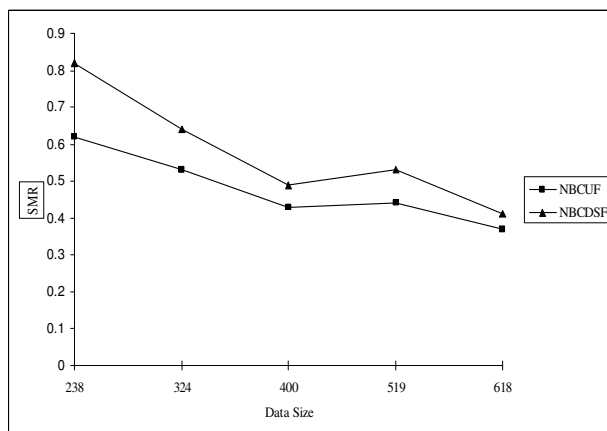


Figure 8. Spam Misclassification Rate results

Detecting spam web pages is one of the major challenges that face search engines in their queries results. Search engines should return high quality results in response to the user's queries. Many search engines necessitate an integration of a healthy detection spam to eliminate all web pages that effect in page ranking algorithm. Several content-based and machine learning techniques were proposed to detect spam pages.

This paper proposed a Naïve Bayes approach that gives weight for user's feedback to improve the training process of the classifier and that consider the existence of some domain specific features that contribute strongly to the spam discrimination, that is, there existence provides evidence that the webpage is spam. This approach proposed mainly to function in the server-side, to reduce the overhead associated with spam pages in the web server.

Our work involved Naïve Bayes classifier in discovering non required pages. The experimental results showed that Naïve Bayes classifier provides on average accuracy equal to 80.2%. For future works we will develop an application as plug-in in one of the open source browser to work as detector in online website pages to notify the users for spam pages currently working on.

Our future work will be to optimize the performance of our Naïve Bayes Classifier by taking into consideration the word-position of the domain specific features, which will contribute to a better accuracy. In addition, the improvement will include the user-oriented feedback model to satisfy the needs of much more users.

References

- [1] S.Atkins, "Size and Cost of the Problem", *In the Proceedings of the Fifty-sixth internet Engineering Task Force(IETF) Meeting*, San Francisco, CA, USA, 2003.
- [2] R.Baeza-Yates and B.Ribeiro-Neto, *Modern Informaion Retrieval*, Edinburgh Gate, England, 1999.
- [3] L.Becchetti, C.Castillo, D.Donato, R.Baeza-Yates, S.Leonardi, "Link Analysis for Web Spam Detection", *ACM Trans. Web*, 2 (1), pp. 1-42, 2008.
- [4] C.Castillo, D.Donato, L.Becchetti1, P.Boldi, S.Leonardi, M.Santini and S.Vigna, "A reference Collection for Web Spam", *SIGIR Forum*, (40), pp. 11-24, 2006.
- [5] Z.Gyöngyi and H.Garcia-Molina, "Web Spam Taxonomy", *In the Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web*, Stanford University, May 2005.
- [6] J.Han, and M.Kamber, *Data Mining: Concept and Techniques*, Morgan-Kaufman, New York, 2000.
- [7] I.Koprinska, J.Poon, J.Clark, and J.Chan, "Learning to Classify E-mail", *Information Science*, 10(177), pp. 2167-2187, 2007.
- [8] C.Lai, "An Empirical Study of Three Machine Learning Methods for Spam Filtering", *Knowledge-Based Systems*, 3 (20), pp. 249-254, 2007.
- [9] C.Lee, Y.Kim, and P.Rhee "Web Personalization Expert with Combining Collaborative Filtering and

Association Rule Mining Technique", *Expert Systems with Applications*, 3(21) pp. 131-137, 2001.

- [10] J.María, G.Cajigas and E.Puertas, "Content Based SMS Spam Filtering", *In the Proceedings of the 2006 ACM symposium on Document engineering*, ACM, 2006.
- [11] A.Ntoulas, M.Najork, M.Manasse, and D.Fetterly, "Detecting Spam Web Pages Through Content Analysis", *In the Proceedings of the 15th international conference on World Wide Web*, ACM, 2006.
- [12] G.Paul, "Better Bayesian Filtering". *In the Proceedings of the 2003 spam conference*, Jan 2003.
- [13] M.Sahami, S.Dunmais, D.Heckerman, and E.Horvitz, "A Bayesian Approach to Filtering Junk E-mail", AAI Workshop on Learning for Text Categorization, July 1998, Madison, Wisconsin, AAI Technical Report WS-1998.
- [14] J.Su and H.Zhang, "Full Bayesian Network Classifiers", *In the Proceedings of 23 rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.
- [15] B.Yu and Z.Xu, "A Comparative Study for Content-Based Dynamic Spam Classification Using Four Machine Learning Algorithms", *Knowledge-Based Systems*, 4(21), pp. 355-362, 2008.
- [16] H.Zhang and D.Li, "Naïve Bayes Text Classifier", *In the Proceedings of the IEEE International Conference on Granular Computing*, 2007.
- [17] L.Zhang, J.Zhu, and T.Yao, "An Evaluation of Statistical Spam Filtering Techniques" *ACM Transactions on Asian Language Information Processing*, 4(1.3), pp. 243-269, 2004.

Authors' Biographies



Hassan M. Najadat is an Assistant Professor in the Department of Computer Information Systems in Jordan University of Science and Technology in Irbid, Jordan. His research interests are centered on data mining, machine learning, database systems and he has authored over nine refereed publications in the areas of clustering, classification, association rules, and text mining. He received his PhD in Computer Science from North Dakota State University in Fargo, USA, MS in Computer Science from University of Jordan in Amman, Jordan, and BS in Computer Science from Mut'ah University in Alkarak, Jordan.



Ismail Hmeidi is an Assistant Professor in the Department of Computer Information Systems in Jordan University of Science and Technology in Irbid, Jordan. His research interests are centered on information retrieval, natural language processing, e-learning, and database systems. He has authored over 13 refereed publications in the areas of query expansion, automatic indexing and text categorization. He received his PhD in Computer Science from Illinois Institute of Technology, USA, MS and BS in Computer Science from Eastern Michigan University, U.S.A.