RESEARCH ARTICLE OPEN ACCESS

Content based web spam detection using naive bayes with different feature representation technique

Amit Anand Soni[1], Abhishek Mathur[2]
Research Scholar SATI Engg. Vidisha (M.P.)
Asst. Professor SATI Engg Vidisha (M.P.)

Abstract

Web Spam Detection is the processing to organize the search result according to specified criteria. Most often this refers to the automatic processing of search result, but the term also applies to the automatic classification of search results into ham and spam. Our work also evaluates change in performance by using different representation for the document vector like term frequency (TF), Binary, inverse document frequency (IDF) and TF-IDF. There are various Benchmark Datasets available for researchers related to web spam filtering. There has been significant effort to generate public benchmark datasets for anti- web spam filtering. One of the main concerns is how to protect the privacy of the users whose ham links are included in the datasets.

We perform a statistical analysis of a large collection of WebPages, focusing on spam detection. Dimension reduction is important part of classification because it provides ease to visualize high dimensional data. This work reduce dimension of training data in 2D and full and mapped training and test data in to vector space. There are several classification here we use Naive Bayes classification and train data set with varying different representation and testing perform with different spam ham ratio

Key-Words: - Content spam, keyword count, variety, density and Hidden or invisible text

I. INTRODUCTION

Search engines are widely used tools for effectively exploring information on the Web. One of the core components of a search engine is its ranking function: when a search engine receives a user query, this function determines the order of presentation of retrieved results (documents or web URLs). The main goal of the ranking process is to promote high-quality and relevant content to the top of the result list, which is an important and challenging problem by itself. In this work we propose a method for improving the quality of ranking of search results that addresses the two important aspects mentioned above through the temporal analysis of search logs.

First, we identify an interesting link between email spam and Web spam, and we use this link to propose a novel technique for extracting large Web spam samples from the Web. Then, we present the Webb Spam Corpus – a first-of-its-kind, large-scale, and publicly available Web spam data set that was created using our automated Web spam collection method.

While performing our classifier evaluations, we identified a clear tension between spam producers and information consumers. Spam producers are constantly evolving their technique to ensure their spam messages are delivered, and information consumers are constantly evolving their countermeasures to ensure they don't receive spam messages. Based on the results of our evolutionary study, we began to question the validity of retraining as a solution for camouflaged messages. Since spammers continually evolve their techniques, we believed they would also evolve their camouflaged messages, making them more sophisticated over time. This process continues until both parties are firmly entrenched in a spam arms race. Fortunately, in this thesis, we propose two solutions that allow information consumers to break free of this arms race.

The second contribution of this thesis is a framework for collecting, analyzing, and classifying examples of Spam attacks in the World Wide Web. Just as email spam has negatively impacted the user messaging experience, the rise of Web spam is threatening to severely degrade the quality of information on the World Wide Web. Fundamentally, Web spam is designed to pollute search engines and corrupt the user experience by driving traffic to particular spammed Web pages, regardless of the merits of those pages. Hence, we present various techniques for automatically identifying and removing these pages from the Web.

II. RELATEDWORK

In this section, we provide an overview of previous efforts to improve the ranking of search results by introducing a better ranking function or a method to detect and eliminate adversarial content, the two major research directions, highly relevant to the present work. The learning-to-rank approaches are capable of combining different kinds of features to train the ranking function. A number of previous works have also focused on exploring the methods to obtain useful information from click-through data, which could benefit search relevance

2.1 Statistical Classification of Email Spam

Email classification can be characterized as the problem of assigning a boolean value ("spam" or "legitimate") to each email message M in a collection of email messages M. More formally, the task of spam classification is to approximate the unknown target function Φ: M! {Spam, legitimate}, which describes how messages are to be classified, by means of a function Ǿ: M! {Spam, legitimate} called the classifier (or model), such that Φ and Ǿ coincide as much as possible.

Different learning methods have been explored by the research community for building spam classifiers (also called spam filters). In our email spam experiments, we focus on three learning algorithms: Na¨ıve Bayes, Support Vector Machines (SVM), and LogitBoost. In the following sections, we will briefly summarize the important details of each of these algorithms.

2.1.1 Naive Bayes

Naive Bayes is one of the simplest classification methods in machine learning. This work use NB because of it takes less training time and Very easy to deal with missing attributes. In the experiments each message is represented as a vector Vi= {T1. . .Tm}(Vi is a feature vector of document i) where T1. . .Tm are the feature and Wi1, Wi2.....Wim are the weight of term T1. . .Tm. We are doing spam filtering in which we have only two classes.

Given a classification task of 2 classes C1, C2 and an unknown pattern, which is represented by a feature vector V, form the two conditional probabilities p (Ci/V) for i=1, 2 Sometimes, these are also referred to as a posteriori probabilities. In words, each of them represents the probability that the unknown pattern belongs to the respective class Ci. Let C1 (spam), C2 (ham) be the two classes in which message belong. Assume that the a priori probabilities P (C1), P (C2) are known. If P (C1), P (C2) are unknown than easily calculated from training dataset. If N total number of mails (spam ham) in training dataset in which N1 belongs to C1 (spam) class and N2 belongs to C2 (ham) class then

$$p(C1) \approx \frac{N1}{N}$$
$$p(C2) \approx \frac{N2}{N}$$

Now compute conditional probability.

$$\mathrm{p}\ (\mathrm{Ci/V}) = \frac{\mathrm{p(Ci)} * \mathrm{p}\ (\mathrm{V/Ci}\ )}{\mathrm{p(V)}}$$

Where p (V) is the pdf of V

$$\mathrm{p(V)} = \sum_{\mathrm{i=1}}^{2} \mathrm{p(Ci)} * \mathrm{p}\ (\mathrm{V/\ Ci}\ )$$

The Bayes classification rule can now be stated as

If p (C1/V) > p (C2/V), V is classified to C1
If p (C1/V) < p (C2/V), V is classified to C2

In case of both are equal then we assign vector X in either class.

$$\mathrm{p(C1)} * \mathrm{p}\ (\mathrm{V/C1}\ ) \lesseqgtr \mathrm{p(C2)} * \mathrm{p}\ (\mathrm{V/C2}\ )$$

Here we don't consider p (V), because it is same for all classes. If the a priori probabilities are equal

$$\mathrm{p(C1)} = \mathrm{p(C2)} = \frac{1}{2}$$

Than

$$\mathrm{p}\ (\mathrm{V/C1}\ ) \lesseqgtr \mathrm{p}\ (\mathrm{V/C2}\ )$$

2.1.2 Dimension reduction:

DR is important part of classification because it provides ease to visualize high dimensional data.

Singular Value Decomposition (SVD):

Data set representation in the form of term document matrix that represents **n** number of document and **m** number of term that describe every document. Suppose A is a document term matrix of **nxm** matrix of data set **A**, A_{ij} shows the feature **j** for documents **i.** Every row of A represented by document (vector of term with m dimension) and number of column called dimension of vector.

Mathematical decomposition of matrix:

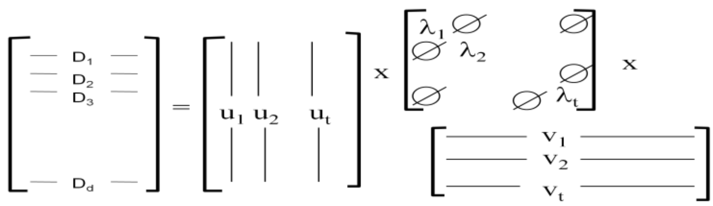
Mathematically matrix A of nxm is decomposing into three parts. Decomposition of matrix is given below.

Here,

d: Represent number of document.

t: Represent number of term in document vector.

$$\mathbf{A}_{[d \times t]} = \mathbf{U}_{[d \times t]} * \mathbf{S}_{[t \times t]} * (\mathbf{V}_{[t \times t]})^{\mathbf{T}}$$



Decomposition of matrix using SVD

Preprocessing of Dataset:

The data set is subjected to the pre-processing. The dataset contains two labeled files which show that the link is spam or normal. From these files constructed our data. Link belongs to which category known to us so it can be easily separable. Wrote a program to extract the content of the pages and save the result into a corresponding text files. Generate a sparse matrix which contains the observation and features. Observations are rows and features are columns.

Table Train Dataset

Datasets	Training		Spam: Ham ratio	Total
	Spam	Ham		
Dataset1	4496	4361	1:1	8857

Table Test Dataset

Datasets	Testing		Spam: Ham ratio	Total
	Spam	Ham		
Dataset1	4500	4500	1:1	9000
Dataset2	3675	1500	2:1	5175
Dataset3	4500	1500	3:1	6000

Feature Representation:
A feature is a word that present in document. Any word in document is called feature if it is satisfies some predefine constraint (feature selection method), Term actually a word refers by **T**; **V** is a feature vector that is composed of the various term formed by analyzing the documents. Every webpage represent by vector. There is various ways to represent vector weight (value of each feature in a vector), vector weight refer by **W**
Some of them given below:

Term Frequency (TF): Term frequency $tf_{i\ j}$ is the number of occurrences of term t_j in document D_i
Note: Different author and research paper used different definition of **TF** some of given below

$$f(tf_{i\,j}) = tf_{i\,j}$$
$$f(tf_{i\,j}) = tf_{i\,j} / l(Di)$$

Where ***l(Di)*** is the length of document ***Di*** **,**means total number of term occurrences in document ***Di***

$$f(tf_{i\,j}) = \sqrt{tf_{i\,j}}$$
$$f(tf_{i\,j}) = 1+\log(tf_{i\,j})$$

We can say that tern frequency refers as a local and I am using TF using

$$f(tf_{i\,j}) = tf_{i\,j}$$

Binary: Binary representation which indicates whether a particular term ***tj*** occurs in a particular document or not. In this representation weight of term ***tj*** define as
Wij=1 if tj ∈ *Di*
Otherwise Wij=0

Document Frequency (DF): Document Frequency df_j is the number of documents in the collection (*Di* where 1≤i≤n) that term **Tj** occurs in. Document Frequency refers as global. In DF we consider only term occurs or not ignore whatever value of **Wij** hold.

Inverse Document Frequency (IDF): Inverse Document Frequency idf_j calculate as follow

$$idf_j=\log(N/df_j)$$

N: Total number of document

Term frequency–Inverse document frequency (TF-IDF): Term frequency multiply by inverse document frequency is called **TF-IDF.**

$$(tf\text{-}idf)_{i\,j} = tf_{i\,j} * idf_j$$

III. Performance Measure

Confusion Matrix for Spam and Ham class

		predicted class	
		ham (-1)	spam (+1)
Actual Class	ham (-1)	TN	FP
	Spam (+1)	FN	TP

- **True positive (TP)**: Correct classifications, spam documents (positive class) classified as spam (positive class)
- **True negative (TN):** Correct classifications, ham documents (negative class) classified as ham (negative class)
- **False positive (FP):** Incorrect classification, FP occurs when the outcome is incorrectly predicted as spam (or positive) when it is actually ham (negative).
- **False negative (FN):** Incorrect classification, FN occurs when the outcome is incorrectly predicted as ham (or negative) when it is actually spam (positive).
- **Accuracy (AC):** accuracy is ratio of correct classification and total number of predictions

$$\textbf{Accuracy} = \frac{\mathbf{TN + TP}}{\mathbf{TN + FP + FN + TP}}$$

Precision:
Precision for a class is the ratio of true class (same class in actual belong to same class in prediction) and total number of item belong for that class in prediction. In other word we can say precision is accuracy of our classification for this class.

$$\text{Precision for spam documentss} = \frac{\mathbf{TP}}{\mathbf{FP + TP}}$$
$$\text{Precision for ham documentss} = \frac{\mathbf{TN}}{\mathbf{FN + TN}}$$

Recall:
Recall for a class is the ratio of true class (same class in actual belong to same class in prediction) and total number of item belong for this class in actual. In other word recall is completeness our classification for this class.

		predicted class	
		ham (-1)	spam (+1)
Actual Class	ham (-1)	150	34
	Spam (+1)	45	120

$$\text{Recall for spam documentss} = \frac{\mathbf{TP}}{\mathbf{FN+TP}}$$

$$\text{Recall for ham documentss} = \frac{\mathbf{TN}}{\mathbf{FP+TN}}$$

False alarm rate:

False alarm rate is define as

$$\text{False alarm rate} = \frac{\mathbf{FP}}{\mathbf{FP+TN}}$$

Or

FAR=1- Recall for ham documents

Ex:

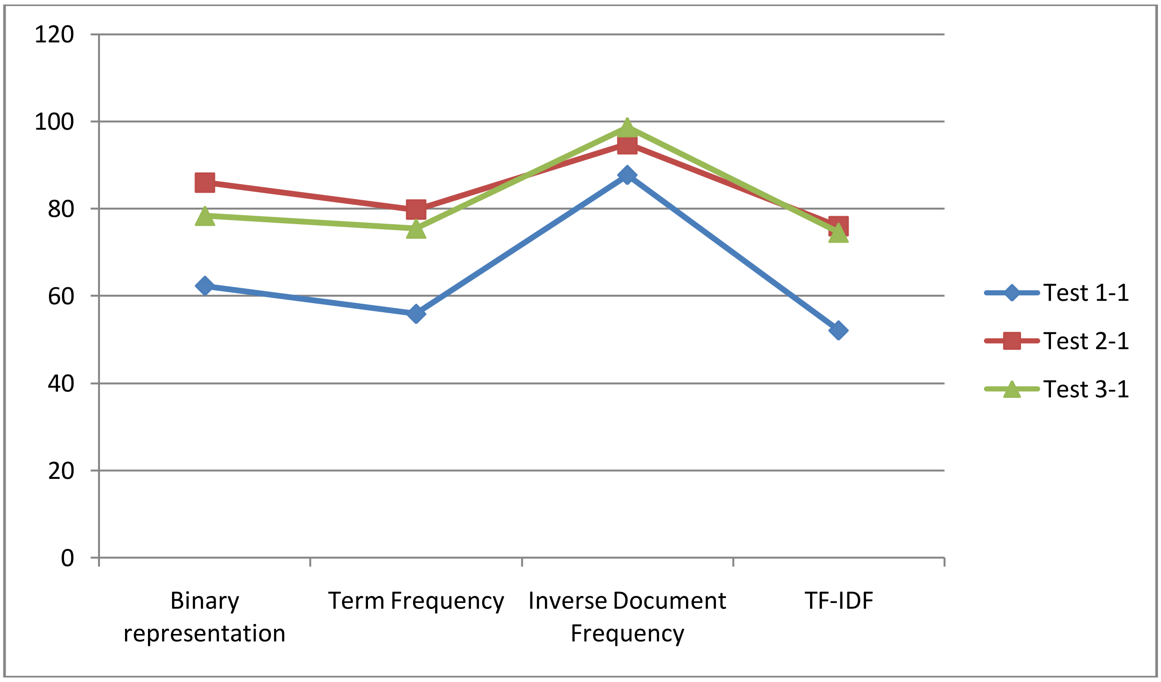
TN:-150, **FP:**-34, **FN:**-45, **TP:**-120
Total ham documents =150+34=184
Total spam documents =45+120=165

Ham documents predicted=150+45=195
Spam documents predicted=120+34=154

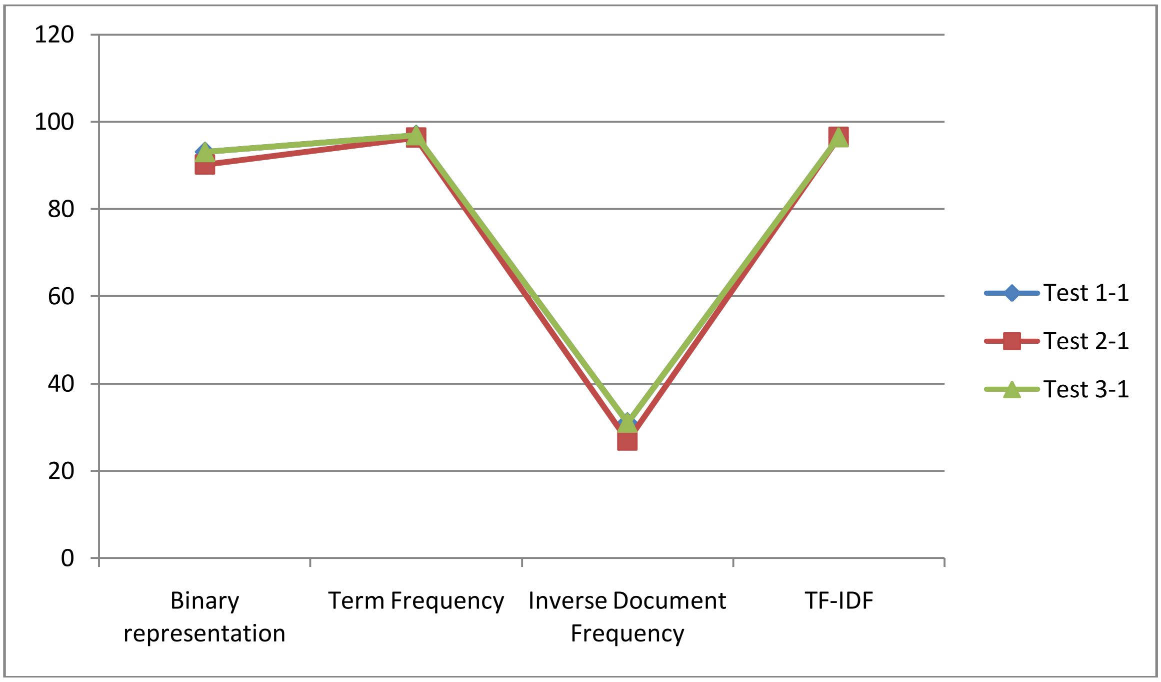
IV. Experimental Results

To determine our filter's performance when it is trained with the various training sets, we evaluate the filter's false positive and false negative rates.

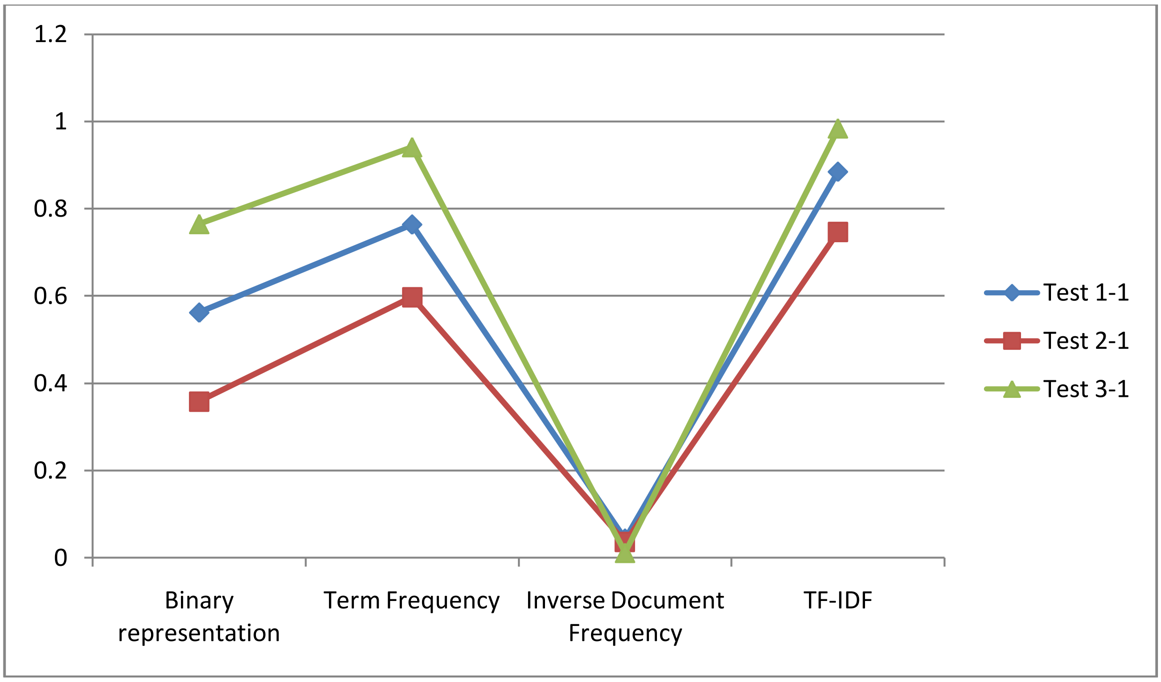
4.1



Spam-precision
4.2

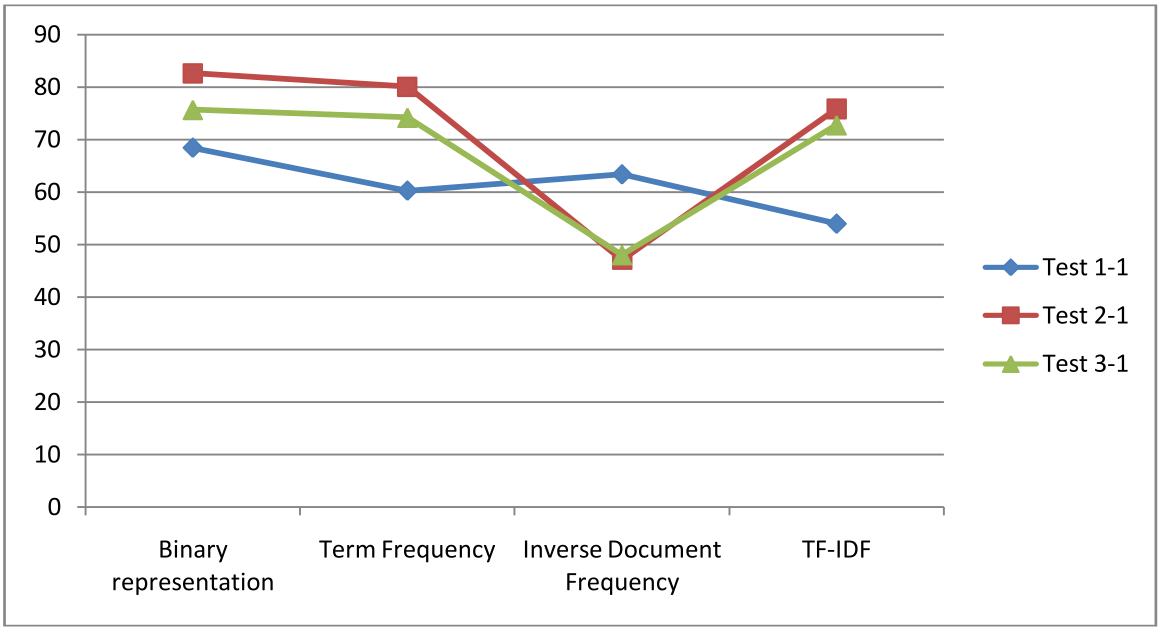


Spam-Recall
4.3



FAR(false alarm rate)

4.4



Accuracy

V. Results and Discussion

Result with Binary representation

Train	Factor	Test 1-1				Test 2-1				Test 3-1			
		Spam			Ham	Spam			Ham	Spam			Ham
Train 1-1		Pre/rec	FAR	ACC	Pre/rec	Pre/rec	FAR	ACC	Pre/rec	Pre/rec	FAR	ACC	Pre/rec
	2	62.35/93.04	0.562	68.43	86.3/43.82	86.05/90.15	0.358	82.63	72.68/64.2	78.48/93.04	0.765	75.65	52.93/23.47
	Full	54.32/90.04	0.757	57.16	70.91/24.27	74.29/79.1	0.671	65.72	39.14/32.93	75.02/90.04	0.899	70.05	25.21/10.07

Result with Inverse Document Frequency

Train	Factor	Test 1-1				Test 2-1				Test 3-1			
		Spam			Ham	Spam			Ham	Spam			Ham
Train 1-1		Pre/rec	FAR	ACC	Pre/rec	Pre/rec	FAR	ACC	Pre/rec	Pre/rec	FAR	ACC	Pre/rec
	2	87.7/31.07	0.0436	63.36	58.12/95.64	94.83/26.97	0.036	47.09	35.01/96.4	98.73/31.07	0.012	48	32.33/98.8
	Full	55.09/87.13	0.71	58.04	69.23/28.96	75.72/77.63	0.61	66.43	41.58/39	74.53/87.13	0.893	68.02	21.65/10.67

Result with Term Frequency

Train	Factor	Test 1-1				Test 2-1				Test 3-1			
		Spam			Ham	Spam			Ham	Spam			Ham
Tr		Pre/rec	FAR	ACC	Pre/rec	Pre/rec	FAR	ACC	Pre/rec	Pre/rec	FAR	ACC	Pre/rec

	2	55.92/96.91	0.764	60.26	88.43/23.6	79.81/96.35	0.597	80.1	81.84/40.27	75.55/96.91	0.941	74.17	39.04/5.933
	Full	56.57/88.49	0.679	60.28	73.58/32.07	76.41/80.27	0.607	68.39	44.82/39.27	76.98/88.49	0.794	71.52	37.36/20.6

Result with TF-IDF

Train	**Factor**	**Test 1-1**				**Test 2-1**				**Test 3-1**			
		Spam			**Ham**	**Spam**			**Ham**	**Spam**			**Ham**
Train 1-1		**Pre/rec**	**FAR**	**ACC**	**Pre/rec**	**Pre/rec**	**FAR**	**ACC**	**Pre/rec**	**Pre/rec**	**FAR**	**ACC**	**Pre/rec**
	2	52.14/96.44	0.885	53.97	76.37/11.49	76/96.49	0.747	75.86	74.66/25.33	74.62/96.44	0.984	72.73	13.04/1.6
	Full	56.46/88.91	0.686	60.17	73.92/31.42	76.09/80.6	0.621	68.23	44.38/37.93	77.31/88.91	0.783	72.12	39.52/21.73

VI. Conclusion

- In Binary representation test data set test 2:1 perform well in terms of recall precision and false alarm rate
- IDF representation gives highest false alarm rate and precision in all testing datasets.
- Data set test 1:1 give less precision in compare to test 2:1 and test 3:1 data set.
- Dimension reduction of training and test data set in to 2D and full 2D perform well as compare to full Dimension.

SUMMARY

The creation of the Internet has fundamentally changed the way we communicate, conduct business, and interact with the world around us. The World Wide Web, and social networking communities, which provide information consumers with an unprecedented amount of freely available information. However, the openness of these environments has also made them vulnerable to a new class of attacks called Spam attacks. Attackers launch these attacks by deliberately inserting low quality information into information-rich environments to promote that information or to deny access to high quality information. These attacks directly threaten the usefulness and dependability of online information-rich environments, and as a result, an important research question is how to automatically identify and remove this low quality information from these environments. In this research paper, we focus on answering this important question by countering Spam attacks in three of the most important information-rich environments: email systems, the World Wide Web, and social networking communities. For each environment, we perform large-scale data collection and analysis operations to create massive corpora of low and high quality information. Then, we use our collections to identify characteristics that uniquely distinguish examples of low and high quality information. Finally, we use our characterizations to create techniques that automatically detect and remove low quality information from online information-rich environments.

References

[1] Alexa, "Alexa top 500 sites." http://www.alexa.com/site/ds/top_sites?ts_mode=global, 2008.

[2] Anderson, D. S. and others, "Spamscatter: Characterizing Internet Scam Hosting Infrastructure," in Proceedings of 16th Usenix Security Symposium (Security '07), 2007

[3] Acquisti, A. and Gross, R., "Imagined communities: Awareness, information sharing, and privacy on the facebook," in Proceedings of the 6th Workshop on Privacy Enhancing Technologies (PET '06), pp. 36 – 58, 2006.

[4] Associated Press, "Official sues students over myspace page." http: //www.sfgate.com/cgi-bin/article.cgi?file=/news/archive/2006/09/22 % /national/a092749D95.DTL, 2006

[5] Androutsopoulos, I., Paliouras, G., and Michelakis, E., "Learning to Filter Unsolicited Commercial E-mail," Tech. Rep. 2004/2, National Center for Scientific Research "Demokritos", 2004.

[6] Amitay, E. and others, "The Connectivity Sonar: Detecting Site Functionality by Structural Patterns," in Proceedings of the

14th ACM Conference on Hypertext and Hypermedia (HYPERTEXT ’03), pp. 38–47, 2003.

[7] Ahamad, M. and others, “Guarding the Next Internet Frontier: Countering Denial of Information Attacks,” in Proceedings of the New Security Paradigms Workshop (NSPW ’02), 2002..

[8] Androutsopoulos, I. and others, “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, pp. 9–17, 2000.

[9] Androutsopoulos, I. and others, “An Experimental Comparison of Naïve Bayesian and Keyword-based Anti-spam Filtering with Encrypted Personal E-mail Messages,” in Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 160–167, 2000.

[10] Androutsopoulos, I. and others, “Learning to Filter Spam E-mail: A Comparison of a Naive Bayesian and a Memory-based Approach,” in Proceedings of the Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases, pp. 1–13, 2000.

[11] Apte, C., Damerau, F., and Weiss, S. M., “Automated Learning of Decision Rules for Text Categorization,” Information Systems, vol. 12, no. 3, pp. 233–251, 1994.