A Survey On: E-mail Spam Messages and Bayesian Approach for Spam Filtering

1P.U.Anitha, 2Dr.C.V.Guru Rao, 3T. Sireesha

1Assistant Professor (CSE), Koneru Lakshmaiah Educational Foundation(KL University), Vaddeswaram, Guntur, India.
2Professor & HOD (CSE), S.R Engineering College, Warangal, India
3Assistant Professor (CSE), Koneru Lakshmaiah Educational Foundation(KL University), Vaddeswaram, Guntur, India

Abstract:

Email has become one of the fastest and most economical forms of communication. However, the increase of email users has resulted in the dramatic increase of spam emails during the past few years. As spammers always try to find a way to evade existing filters, new filters need to be developed to catch spam. Generally, the main tool for email filtering is based on text classification. A classifier then is a system that classifies incoming messages as spam or legitimate(ham) using classification methods. The most important methods of classification utilize machine learning techniques. In this paper we compared some machine learning techniques and we discussed about Bayesian Algorithm for Spam detection with implementation results.

Key words— Spam; Ham, E-mail Filtering; SVM, HMM, Bayesian Algorithm;

1. Introduction to E-mail Filtering:

The main annoying effects of spam are decreasing productivity of employees, wasting valuable storage on mail-servers, harming Internet traffic and increasing possible information loss depending on filtering policies.

It is rapid information exchange Era and one of the advances, secure, cheap, reliable and fast technologies for information exchange is Email. Users of Emails are increasing day by day and also increasing the volume of unwanted mails (spam). Also popular medium of communication for E – Commerce is Email which has opened the door for direct marketers to bombard the mails which fills the mail boxes of users with unwanted mails and as same copy of mail is there on many users mailbox on same server it is just wastage of resource and also waste of bandwidth. Spam mail is also called as unsolicited bulk mail or junk, so we say spam Email is unwanted internet Email. Spam is an ever-increasing problem. The number of spam mails is increasing daily – studies show that over 90% of all current email is spam. Added to this, spammers are becoming more sophisticated and are constantly managing to outsmart ‘static’ methods of fighting spam. The techniques currently used by most anti-spam software are static, meaning that it do this, spammers simply examine the latest anti spam software and then check whether a particular word appears in the mail. High probability indicates the new e-mail as spam e-mail[2].

1.1 Bayesian Approach

Naive Bayesian is a fundamental statistical approach based on probability initially proposed by Sahami et al. (1992). [2] The Bayesian algorithm predicts the classification of new e-mail by identifying an e-mail as spam or legitimate.[2] This is achieved by looking at the features using a ‘training set’ which has already been pre-classified correctly and then checking whether a particular word appears in the mail. High probability indicates the new e-mail as spam e-mail[2].
1.2 Automatic classification:

A classifier is an algorithm that processes a linguistic input and assigns it a class from a user-defined set. It usually denotes a statistical model induced through machine learning. The algorithm works off a set of weighted contextual features.

Table 1: Classifier examples

<table>
<thead>
<tr>
<th>Example</th>
<th>Unit of linguistic input</th>
<th>What type of classes</th>
<th>Class labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS-tagging a text</td>
<td>word</td>
<td>Part-of-speech</td>
<td>Noun, Verb, Adj, Det...</td>
</tr>
<tr>
<td>Grammer Error Checker</td>
<td>Sentence</td>
<td>Grammaticality</td>
<td>Gramatical/Not</td>
</tr>
<tr>
<td>Pronoun resolution</td>
<td>NP + pronoun</td>
<td>Co-reference</td>
<td>co-refers/not</td>
</tr>
<tr>
<td>Spam filter</td>
<td>document (email)</td>
<td>Spam or not</td>
<td>Spam/Ham</td>
</tr>
<tr>
<td>Language identifier</td>
<td>document</td>
<td>Which language</td>
<td>ENG, SPA, FRN, JAP, CHI, KOR, ...</td>
</tr>
<tr>
<td>Topic identifier</td>
<td>document</td>
<td>What topic</td>
<td>Politics, Sports, Entertainment,...</td>
</tr>
<tr>
<td>Automatic essay grader</td>
<td>document (essay)</td>
<td>Quality of writing</td>
<td>5, 4, 3, 2, 1, 0</td>
</tr>
<tr>
<td>Military intelligence</td>
<td>document (message)</td>
<td>Threat assessment</td>
<td>Contains a threat</td>
</tr>
</tbody>
</table>

Document classification is an example of computer science engineering called machine learning. Just like humans learn from "experience", a computer algorithm learns from data. Machine learning is not limited to linguistic data.

1.3 Supervised vs. unsupervised learning:

1.3.1 Supervised:

Training data and test data are pre-labeled (by humans) with desired "correct answers". A statistical model is trained in the training data which maximizes the likelihood of it producing correct labels for the training portion. The trained model can be used to make predictions on unseen data: It is tested on the test data for accuracy.

1.3.2 Unsupervised:

There are no pre-determined categories. Input items are sorted into clusters that share similarities.

Examples:

Find groups of "similar" texts among a set of student-written papers. Given a bunch of texts in a language, find morphemes. How? Assuming grammatical affixes are statistically frequent; induce a set of frequently occurring word partitions.

1.3.3 Supervised machine learning:

Figure 1: Supervised Machine Learning Process

A classification decision must rely on some observable evidence, Deciding what features are relevant. Two types:

Kitchen sink strategy: Throw a set of features to the machine learning algorithm, see what features are given greater weight and what gets ignored.

Eg: using every word in a document as a feature:
Hand-crafted strategy: Utilizing expert knowledge, determine a small set of features that are likely to be relevant. 

E.g.: grammatical error detection. For each sentence, determine, grammatical or ungrammatical.

Hand-coded features: Subject-verb agreement, fragment or not, etc. A classification decision involves reconciling multiple features with different levels of predictive power. Different types of classifiers use different algorithms for:

1. Determining the weights of individual features in order to maximize its labeling success in the training data
2. When given an input, using the feature weights to compute the likelihood of a label.

1.4 Popular machine learning methods:
Decision tree, Maximum entropy (ME), Hidden Markov model (HMM), Support vector machine (SVM), Naïve Bayes.

1.4.1 Decision tree:
The decision tree is one of the most famous tools of decision-making theory. Decision tree is a classifier in the form of a tree structure to show the reasoning process. Each node in decision tree structures is either a leaf node or a decision node. The leaf node indicates the value of the target attribute of instances. The decision node indicates two or more branches and each branch represents values of the attribute tested. When classifying an unknown instance, the unknown instance is routed down the tree according to the values of the attributes in the successive nodes. C4.5 is one of the most popular decision trees algorithms. According to the splitting node strategy, C4.5 builds decision trees from a set of training data. At each node of the tree, C4.5 chooses one attribute that most effectively splits its set of instances into subsets. The C4.5 algorithm recursively visits each decision node and selects the optimal split until no further splits are possible. Fig. 1 is an illustration of the structure of decision tree built by the C4.5.

Fig. 2: An example of decision tree
In Fig. 2, Outlook, Humidity and Wind in inner nodes of the tree are condition attributes and Yes and No are the values of decision attribute in the dataset.

The basic framework of the proposed decision tree ensemble based classification algorithm of spam email is shown in Fig. 3.

Fig. 3: The schematic view of the algorithm
In mathematics, the performances of the above methods are to build judgment surfaces. In fact, this surface is hard to build, because we will encounter
various difficulties. Decision Tree is one of the most commonly method, because it has the following advantages:

a. As Decision Tree is divided into a few steps to carry out; the accuracy of the judgment is higher.

b. Judgment rules can be chosen some simple ones at each step.

c. we needn’t to use all the features at each step. A few effective ones are enough. It will reduce the workload of each step.

d. classify faster.

The disadvantage of the method is that it will cost much time to build classifier. It has another question: the stability of the tree’s structure. In addition, the use of the features and classification methods for different levels is a difficult job.

1.4.2 Support Vector Machine:
Support Vector Machine (SVM) can be expressed as to find a hyper-plane, which can separate the data of the training set (in other words, it has minimum training error) and has a minimum weight of vector. After using vector space model to express the characteristics of the document, the user’s needs information can be seen as a document. That is to say, it can be expressed as a vector. The degree of similarity between documents and user’s interests can be signified by the cosine similarity of the documents and user’s interests.

Given some training data \( D \), a set of \( n \) points of the form:

\[
D = \{(x_i, y_i) \mid x_i \in \mathbb{R}^p, y_i \in \{-1, 1\}\}_{i=1}^n
\]

Where the \( y_i \) is either 1 or -1, indicating the class to which the point \( x_i \) belongs. Each \( x_i \) is a \( p \)-dimensional real vector. We want to find the maximum-margin hyperplane that divides the points having \( y_i = 1 \) from those having \( y_i = -1 \). Any hyperplane can be written as the set of points \( x \) satisfying

\[
w \cdot x - b = 0
\]

If the training data are linearly separable, we can select two hyperplanes in a way that they separate the data and there are no points between them, and then try to maximize their distance. The region bounded by them is called "the margin". These hyperplanes can be described by the equations

\[
w \cdot x - b = 1 \quad \text{and} \quad w \cdot x - b = -1.
\]

Fig4: \( H_4 \) does not separate the classes. \( H_2 \) does, but only with a small margin. \( H_3 \) separates them with the maximum margin.

Fig5 : Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.

SVM has the following features:

a. They have a minimum distance of the training sample from hyper-plane in all kinds of categories.

b. Though it has little vectors, it contains all the information which we used to classifying.

c. Most training samples are not support vector, so the removal or reduction of these samples has no impact on the classifier

When the sample space can be non-linear sharing, the sample space can be mapped onto a high-dimensional space by using kernel function.
Operational point plot can be computed in the new space.

In order to facilitate calculation, we usually choose three kernel functions as mapping functions: polynomial kernel function, RBF and Sigmoid function. SVM is a structure based on risk minimization of the pattern recognition method. When it is used to filtering E-mails, the method aims to find a hyper plane. This plane can clarify E-mails into legitimate E-mails and spam as rightly as possible, and the two types of the data are farthest form the plane. SVM can achieve good results of classification with a small training set. However, SVM can’t adapt changing E-mails, too. Filtering effect is bad, when filtering new spam.

1.4.3 Maximum entropy:
The maximum entropy principle is well-accepted in the statistics community. It states that given a collection of known facts about a probability distribution, choose a model for this distribution that is consistent with all the facts but otherwise is as uniform as possible. Hence, the chosen model does not assume any independence between its parameters that is not reflected in the given facts.

Advantages of Maximum Entropy Method:
1. The data sparseness problems in language modeling can be avoided by setting only reliable marginal constraints.

2. The Maximum Entropy Framework has the ability to combine different kinds of statistical dependencies into one unified framework.

3. Creating a “smooth” model that satisfies all empirical constraints.

4. Incorporating various sources of information (e.g. topic and syntax) in a unified language model.

Disadvantages of Maximum Entropy Method:
1. High computational complexity of model parameter estimation procedure.

2. Heavy computation load in using ME models during recognition.

1.4.4 Hidden Markov model (HMM):
A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a hidden Markov model, the state is not directly visible, but output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states. HMM process with unobserved (hidden) states. Hidden Markov models are especially known for their application in temporal pattern recognition such as speech, handwriting, gesture recognition, part-of-speech tagging, score, partial and bioinformatics. A hidden Markov model can be considered a generalization of a mixture model where the hidden variables (or latent variables), which control the mixture component to be selected for each observation, are related through a Markov process rather than independent of each other.

Fig6. Probabilistic parameters of a hidden Markov model
x- States
y- Possible transition observations
a- State transition probabilities
b- Output probabilities

Eg:
Fig 7: A diagram illustrating an HMM and the different ways a,a,b,c can be generated by the HMM.

Probability of an observed sequence:

The task is to compute, given the parameters of the model, the probability of a particular output sequence. This requires summation over all possible state sequences:

The probability of observing a sequence

\[ Y = y(0), y(1), \ldots, y(L-1) \]

of length \( L \) is given by

\[ P(Y) = \sum_{X} P(Y | X) P(X), \]

where the sum runs over all possible hidden-node sequences

\[ X = x(0), x(1), \ldots, x(L-1). \]

A basic, non-Bayesian hidden Markov model can be described as follows:

- \( N \) = number of states
- \( T \) = number of observations
- \( \theta_{i} \) = emission parameter for an observation associated with state \( i \)
- \( \phi_{i,j} \) = probability of transition from state \( i \) to state \( j \)
- \( \phi_{i} \) = \( N \)-dimensional vector, composed of \( \phi_{i,j} \) must sum to 1
- \( x_{t} \) = observation at time \( t \)
- \( y_{t} \) = observation at time \( t \)
- \( F(\theta) \) = probability distribution of an observation, parametrized on \( \theta \)
- \( x_{t} \sim \text{Categorical}(\phi_{i}) \)
- \( y_{t} \sim F(\theta) \)

In a Bayesian setting, all parameters are associated with random variables, as follows:

\[ \begin{align*}
X, T &= \text{as above} \\
\theta_{i} &\sim \text{as above} \\
\phi_{i,j} &\sim \text{as above} \\
\alpha, \beta &\sim \text{as above} \\
\theta &\sim \text{Gaussian} \left( \mu, \sigma \right) \\
\phi_{i} &\sim \text{Symmetric} \left( \frac{1}{N} \right) \\
x_{t} &\sim \text{Categorical}(\phi_{i}) \\
y_{t} &\sim F(\theta)
\end{align*} \]

Advantages of HMMs:

1. HMMs are used because they have proved effective in a number of domains. The most significant of these is speech recognition, where it forms the basis of most commercial systems. They have also proved effective for a number of other tasks, such as handwriting recognition and sign language recognition.

2. One of the most important advantages of HMMs is that they can easily be extended to deal with strong TC tasks. In the training stages, HMMs are dynamically assembled according to the class sequence.

3. When recognition is attempted for class sequence then large HMMs are assembled from the smaller individual models. This is done by converting from a grammar into a graph representation, then replacing each node in the graph with the appropriate model. This process is called "embedded re-estimation". To find out what the class sequence was, the most probable path is calculated. The path traversed corresponds to a sequence of classes, which is our final classification.

4. HMM uses only positive data, they scale well; since new words can be added without affecting learnt HMMs. It is also possible to set up HMMs in such a way that they can learn incrementally.
5. The basic theory of HMMs is also very elegant and easy to understand. This makes it easier to analyze and develop implementations for.

**Disadvantages of HMMs:**

1. They make very large assumptions about the data.

2. The number of parameters that need to be set in an HMM is huge.

3. The amount of data that is required to train an HMM is very large.

4. HMMs only use positive data to train. In other words, HMM training involves maximizing the observed probabilities for examples belonging to a class. But it does not minimize the probability of observation of instances from other classes.

**2. Bayesian Algorithm**

“Bayesian” named after Thomas Bayes who was a famous mathematician, he developed a new possibility of inference theory which can predict the future by the past. Bayesian classification model is a typical method of classification based on statistical models. Bayesian theorem is the most important formula in Bayesian theory, and it is the foundation to learning Bayesian. It combines the priori probability with the posterior probability wonderfully, and it predicts the posterior probability by using the prior probability and the sample data. Anti-spam model based on Bayesian Classifier deals with study of E-mails. It can be used to training E-mails sample set, identifying the division, refining the characters of the spam (training results). We can achieve Bayesian Classifier based on this model, use it to detect E-mail, and filter E-mail, if it is classified as spam. Bayesian filters are based on "self-learning" smart technology, and it can be adapt itself to the new tricks. Bayesian filters provider the legitimate E-mails with protection. Bayesian filtering technology has been applied to more and more anti-spam products, and it is one of the most effective anti-spam technologies. In spam filters, the naive Bayesian has good effect, and it is very simple. An Email $e_i$ can be expressed by $x_{e_i} = (x_1, x_2, \ldots, x_n)$ and the probability of $e_i$ and belonging to $C_i$ (generally Emails can be divided into two types legitimate Email and Spam is:

$$P(C_k | x_{e_i}) = \frac{P(C_k) \prod P(x_j | C_k)}{\sum P(C_k) P(x_{e_i} | C_k)}$$

In the above method, we suppose that one feature is independent of the others, that is to say that the incidence of one word occurring in the E-mail is independent of the incidence of the other word occurring.

**2.1 The Reason to use Bayesian**

Spam filtering and the generally classifications have numerous differences, as follows:

- Both the mail servers and user client, the real-time requirements of the spam filtering is higher. We must choose the text classification which is simpler and faster as far as possible.

- The relation of the general classification technology and text provider is either no matter or cooperative relations. But the relation of the spam manufacturers and filter developers is antagonistic relation. One side wants to filter spam as soon as possible. But the other wants to manufacture much spam and bypass the filters by all means.

- When it comes to the results of the categorization, people do not want to misjudge non-spam for spam. Otherwise, the non-spam will be filtered. It has nothing to do with that spam is misjudged as non-spam.

**Advantages of Bayesian filtering algorithm:**

- Through the study of new spam and normal samples of spam, Bayesian will be able to combat the latest spam, and has better effect on various font.

- Bayesian Algorithm is difficult to deceive.

- In competence (or) efficiency, Bayesian Algorithm is superior to the other algorithms.
- Bayesian Algorithm can update with the constantly receiving E-mails.

Therefore, Bayesian Algorithm is the most common method which is used to filtering spam, in the form of Naive Bayesian classifier.

### 2.2 Improved Naive Bayesian Algorithm:

**Stage 1: Training**
- Parse each email into its constituent tokens
- Generate a probability for each token $W\in\mathcal{W}$
- $S[W] = \frac{c_{spam}(W)}{c_{ham}(W) + c_{spam}(W)}$
- Store spamminess values to a database

**Stage 2: Filtering**
- For each message $M$
  - While (M not end) do
    - Scan message for the next token $T_i$
    - Query the database for spamminess $S(T_i)$
    - Calculate accumulated message probabilities $S[M]$ and $H[M]$
    - Calculate the overall message filtering indication by:
      $I[M] = f(S[M]; H[M])$
    - $f$ is a filter dependent function, such as $I[M] = \frac{1 + S[M] - H[M]}{2}$
    - If $I[M] >$ threshold
      - Msg is marked as spam
    - Else
      - Msg is marked as non-spam

We know that Naïve Bayesian algorithm is based on “Bayesian assumptions” which assume that each of the characteristics is independent by analyzing the theory. In fact, this assumption is difficult to exist. Experimental data also show that Bayesian algorithm makes important information lost, and leads to misjudging spam and legitimate mail.

1. **Total probability Theorem:**
   \[
P(B) = \sum_{i=1}^{N} P(B|A_i)P(A_i)
   \]
2. **Bayes’ Theorem:**
   \[
P(H|X) = \frac{P(X|H)P(H)}{P(X)} = \frac{P(X|H)P(H)}{P(X)}
   \]
3. **Let $X$ be a data sample (“evidence”):** class label is unknown
4. **Let $H$ be a hypothesis that $X$ belongs to class $C$**
5. **Classification is to determine** $P(H|X)$, (i.e., *posteriori probability*): the probability that the hypothesis holds given the observed data sample $X$
6. **$P(H)$ (prior probability):** the initial probability E.g., $X$ will buy computer, not considering of age, income
7. **$P(X)$: probability that sample data is observed**
8. **$P(X|H)$ (likelihood):** the probability of observing the sample $X$, given that the hypothesis holds Ex., Given that $X$ will buy computer, the prob. that $X$ is 31..40, medium income

### 2.3 Prediction Based on Bayes’ Theorem:

Given training data $X$, *posteriori probability of a hypothesis $H$*, $P(H|X)$, follows the Bayes’ theorem

\[
P(H|X) = \frac{P(X|H)P(H)}{P(X)} = \frac{P(X|H)P(H)}{P(X)}
\]

Informally, this can be viewed as posteriori = likelihood x prior/evidence

Predict $X$ belongs to $C_i$ iff the probability $P(C_i|X)$ is the highest among all the $P(C_k|X)$ for all the $k$ classes

Practice difficulty: It requires initial knowledge of many probabilities, involving significant computational cost.

### 2.4 Bayesian Spam Filtering:

Bayesian spam filtering can be conceptualized into the model presented in Figure 1. It consists of four major modules, each responsible for four different processes: message tokenization, probability estimation, feature selection and Naive Bayesian classification. When a message arrives, it is firstly tokenized into a set of features (tokens), $F$. Every feature is assigned an estimated probability that indicates its spaminess. To reduce the dimensionality of the feature vector, a feature selection algorithm is applied to output a subset of the features. The Naive Bayesian classifier combines the probabilities of every feature in $F$, and estimates the probability of the message being spam. In the following text, the process of Naive Bayesian classification is described, followed by details concerning the measuring performance. This order of explanation is necessary because the sections concerned with the first three
modules require understanding of the classification process and the parameters used to evaluate its improvement.

Incoming text (e-mail)

Message

Probability

Feature Selection

Bayesian Classifier

Remove Message

Process Message

Fig 7: Classification of text messages

3. Anti-Spam Approach:

Bayesian filtering [20] works on the principle that the probability of an event occurring in the future can be inferred from the previous occurrences of that event. Spam emails can be processed through Bayesian filters using keywords, as widely known. Single keyword or multiple keyword combinations can be used. Along with the keywords, we propose to use keyword contexts or contexts, in short. Making a spam decision by merely using keywords cannot be that accurate. Once the keyword is checked using a context, the picture becomes clearer and a more accurate decision can be taken. Context is a set of remaining keywords that is mapped to every keyword chosen as shown in figure 1. For example, if the [keyword 1] has a context of [keyword 2, keyword 3 … keyword n], then [keyword 2] has a context of [keyword 1, keyword 3 … keyword n] etc. Generally, the keywords chosen can be uncommon or critical nouns (or combinations), along with acronyms, names etc. An exemption file list can be used during implementation. The anti-spam algorithm can be described as follows. Accept the incoming mails and extract keywords from subject line and email contents as one-keyword (k1i), two keyword (k2i), three-keyword (k3i) or multi keyword sets. Form contexts Cij for content keywords (k1i), two-keyword(k2i) and three-keyword (k3i) sets. The context for any keyword is a set that contains all other keywords except itself. Thus a keyword or keyword combinations can have more than one context, as different spam can contain different sets of keyword combinations. Use the identified keywords to assign a Bayesian probability related score. The keyword contexts are compared to the set of existing keywords, to find a context matching percent (CMP). Three approaches are discussed here – Bayesian using single keywords, Improved Bayesian with multiple keywords and Improved Bayesian with keyword context matching.

3.1 Bayesian Approach with Single keywords

The Bayesian probability $p(k)$ for keyword $k$ is given as in equation 1:

$$P(k) = \frac{s(k)}{s(k) + ns(k)}$$

where, $s(k)$ is the number of spam emails with keyword $k$ and $ns(k)$ is the number of non-spam emails with keyword $k$. The overall weighted spam score is calculated as follows. The Bayesian score for single keywords and multi-keywords are calculated and no weights are assigned to multi-keywords. The keyword scores are totaled to get the spam score for a given mail. The Bayesian probability $p(sk)$ for single keyword set $sk$,

$$p(sk) = \frac{s(sk)}{s(sk) + ns(sk)}$$

where, $s(sk)$ is the number of spam emails with all single keyword set $sk$ and $ns(sk)$ is the number of non-spam emails with all single keyword set $sk$. Similar approach is adopted for multi-keywords.

3.2 Improved Bayesian Approach with Multiple keywords

In comparison to the previous method[12], here weights are assigned to multiple keywords. Weights associated with one, two and three keywords (or multiple keywords) are $Wk1i$, $Wk2i$ and $Wk3i$ respectively, where $i = 1$ to $n$ (where $Wk1i < Wk2i < Wk3i$). Spam score for one, two and three keywords are given as $Sk1i$, $Sk2i$ and $Sk3i$ respectively, where $i = 1$ to $n$. Bayesian calculation is done with weights and keywords scores are determined, which are
eventually added to get the spam score. The Bayesian probability \( p(mk) \) for multi-keyword set \( mk \),

\[
p(mk) = \frac{s(mk)}{s(mk) + ns(mk)}
\]

where, \( s(mk) \) is the number of spam emails with all multi-keyword set \( mk \) and \( ns(mk) \) is the number of non-spam emails with all multi-keyword set \( mk \). In the simulation done, the multiple keywords present are assigned different weights in spam score calculation as follows: Two keywords are assigned a weight of \( MK\_WEIGHT \) (constant value), three keywords are assigned a weight of \( MK\_WEIGHT \times 3 \), four keywords or more are assigned a weight of \( MK\_WEIGHT \times 4 \). Single keywords are not assigned any weights.

3.3 Bayesian with Keyword-Context Approach:

To further improve the Improved accuracy, we add the keyword context score to the improved Bayesian score. Spam score for one, two and three keywords with corresponding keyword contexts are \( Skc1i \), \( Skc2i \) and \( Skc3i \) respectively, where \( i = 1 \) to \( n \). This score is calculated with respect to the matches spam mail keywords contexts find in the existing database of keywords. For example, consider a keyword [viagra] that has a context of [word1, word2, word3, word4] in a mail received. Matching percentage can be given as \( x\% \) for keyword context match. If two words match out of four, then matching percentage would be 50%. The keyword context score \( (Skcij) \) would be a function of this matching percentage. This spam score for keyword-context pairs can have a greater contribution in the overall score. This is effected by \( W1 \) and \( W2 \), where \( W1 \) is the weight (say, 70\%) associated with keyword score and \( W2 \) (say, 30\%) is associated with keyword-context score component in equation 4. These values can be fine-tuned for best results. Weights associated with contexts that corresponds to one, two and three keywords are \( Wkc1i \), \( Wkc2i \) and \( Wkc3i \) respectively, where \( i = 1 \) to \( n \) (where \( Wkc1i < Wkc2i < Wkc3i \)). The Total Spam Score = Total weighted Bayesian score for all keywords found + Total weighted score based on matching percent for all keyword-contexts found, corresponding to all keywords. That can be mathematically expressed as in equation 4:

\[
S_{\text{total}} = \sum_{i=1;j=1}^{i=n;j=n} W1(Sk_{ij} \times W_{ki}) + W2(Skc_{ij} \times W_{kc_{ij}})
\]

For each keyword, the corresponding contexts are formed. The presence of spam keyword itself doesn’t guarantee a good spam score, but keywords with contexts if present, can give a good spam score. Threshold and weight factors should be fine tuned in different stages.

4. Actual Implementation

We divided this implementation into following three parts.

A. Training
B. Classification

A. Training
In Training part we have to train following three database of Spam Filter.

• Origin Email id with counter (Blacklist).
• Spam with counter.
• Legitimate with counter.

For our system we have used some mails from following E-mail ID to train the database.

• govindchris@gmail.com
• anitha_podishetty@yahoo.co.in

In this algorithm we have neglected some common occurring words, list of these words are as below

hi, hello, dear, regards, thank, thanks, of, into, they, she, it, been, he, in, the, how, where, an, out, you, i, am, there, not, can, could, would, will, if, has, have, why, who, had, with, your, or, any, my, we, so, date, to, from, mon, monday, tue, tuesday, wed, wednesday, thu, thursday, fri, friday, sat, saturday, sun, sunday, jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec, let, make, put, seem, take, about, among, at , between, now, out, still, almost, even, much, quite, very, please.

A.1 Training (Algorithm)
1. After classification retrieve sender email id of all spam mail.
2. If sender email id of spam mail is available in origin (blacklist) database then just increase its count, otherwise insert email id in origin (blacklist) database.
3. Retrieve sender email id of all legitimate email.
4. If sender email id of legitimate mail is available in origin (blacklist) database then set value of count is zero.
5. Extract features (word) from all spam mail
6. Update database of spam mail; if word available then increase its count by one otherwise insert it as new word with count one in spam databases.
7. Update database of legitimate mail; if word available then increase its count by one otherwise insert it as new word with count one in legitimate databases.
8. Database improvement is complete.

4.1 Training (Flow Chart)

4.2 Classification Process (Algorithm)

1. Download new mail.
2. Retrieve database or sender email id.
3. If there is no sender id then classify as a spam.
4. If sender email id available in origin database then check its count, if count is greater than 20 then classify this mail is a spam otherwise send this mail in second level (Bayesian) to classify.
5. In second level (Bayesian) receive mail which is not classified by first level.
6. Extract features (word) from all mail and store it in temporary database with frequency of occurrence in same mail.
7. If there is no text in mail then classify as a spam.
8. If there is any attachment then give message to check this mail because filter is not able to read attachment.
9. Calculate probability for spam and legitimate by above Bayesian formula for each word.
10. Store probability of each word for spam and legitimate in temporary database.
11. Calculate sum of probability of all word of same file for spam and legitimate.
12. If sum of probability for spam is greater than legitimate then classify as spam otherwise legitimate.
13. If sum of probability for spam and legitimate is same then classify as legitimate.
14. Classification process is complete.

A.4 Classification Process (Flow Chart)
5. Results

### TABLE 1

<table>
<thead>
<tr>
<th>Total Mail = 28</th>
<th>Spam</th>
<th>Legitimate</th>
<th>Actual Spam</th>
<th>Actual Legitimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesian</td>
<td>17</td>
<td>6</td>
<td>18</td>
<td>5</td>
</tr>
</tbody>
</table>

### TABLE 2

<table>
<thead>
<tr>
<th>Total Mail = 17</th>
<th>Spam</th>
<th>Legitimate</th>
<th>Actual Spam</th>
<th>Actual Legitimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesian</td>
<td>9</td>
<td>4</td>
<td>9</td>
<td>4</td>
</tr>
</tbody>
</table>

From the above tables, we can observe that the Bayesian Approach Improves the Accuracy.

6. Conclusion

In the time of growing problem of Junk Email, we have made a system which classifies junk mail automatically; this system uses the concept of Bayesian theorem for classification task. The efficiency of this kind of system is enhanced by considering not only words of mail as feature but we can consider other domain specific features which provide strong evidence about Junk. Also we can set some manually made handy rules along with system to improve system performance. Here we have not considered header of the mail so in future work we can use header to improve system accuracy.

7. References:


