ABSTRACT

Spam emails are causing major resource wastage by unnecessarily flooding the network links. Though many antispam solutions have been implemented, the Bayesian spam score approach looks quite promising. A proposal for spam detection algorithm is presented and its implementation using Java is discussed, along with its performance test results on two independent spam corpuses – Ling-spam and Enron-spam. We use the Bayesian calculation for single keyword sets and multiple keywords sets, along with its keyword contexts to improve the spam detection and thus to get good accuracy.

KEYWORDS spam mails; bayesian approach; spam corpus; keyword sets, context matching;

INTRODUCTION:

Spam emails are causing major resource wastage by unnecessarily flooding the network links. Though many anti-spam solutions have been implemented, the Bayesian spam score approach looks quite promising. A proposal for spam detection algorithm is presented and its implementation using Java. We use the Bayesian calculation for single keyword sets and multiple keywords sets, along with its keyword contexts to improve the spam detection and thus to get good accuracy. E-mail spam slowly but exponentially grew for several decades to several billion messages a day. Spam has frustrated, confused, and annoyed e-mail users. Laws against spam have been sporadically implemented, with some being opt-out and others requiring opt in e-mail. The total volume of spam (over 100 billion emails per day as of April 2008) has leveled off slightly in recent years, and is no longer growing exponentially. The amount received by most e-mail users has decreased, mostly because of better filtering. About 80% of all spam is sent by fewer than 200 spammers. Botnets, networks of virus-infected computers, are used to send about 80% of spam. The cost of spam is borne mostly by the recipient, so it is a form of postage due advertising. Spam filters help us in our fight against spam and spammers. But there isn’t a perfect filter. An ideal spam filter should generate a false positive and a false negative. Unfortunately this is hard to achieve, quite impossible actually. But it’s great to know that there are email delivery assurances systems which can help online marketers optimize their campaigns and mailing.

About Spam

With the increasing popularity of electronic mail (or e-mail), several people and companies found it an easy way to distribute a massive amount of unsolicited messages to a tremendous number of users at a very low cost. These unwanted bulk messages or junk emails are called spam messages. The majority of spam messages that has been reported recently are unsolicited commercials promoting services and products including sexual enhancers, cheap drugs and herbal supplements, health insurance, travel tickets, hotel reservations, and software products. They can also include offensive content such pornographic images and can be used as well for spreading rumors and other fraudulent advertisements such as make money fast. E-mail spam has continued to increase at a very fast rate over the last couple of years. It has become a major threat for business users, network administrators and even normal users. A study in July 1997 reported that spam messages constituted approximately 10% of the incoming messages to a corporate network.

More recently, Message Labs stated in its 2006 Annual Security Report that spam activity has increased significantly in 2006 with levels that reach 86.2% of the e-mail traffic. The report has also indicated that largely due to the increased sophistication of robot networks, a.k.a. botnets, the spam volumes have increased by 70% over the last quarter of 2006 which in turn increased the overall email traffic by a third. Based on projections of current analysis and trends, it was expected that by the end of 2007, spam will continue to rise, reaching a plateau at around 92% of e-mail traffic. There is a prediction that by year 2015 spam will exceed 95% of all e-mail traffic. Although these figures might not be accurate enough, what can be concluded is that spam volume is dramatically increasing over years. Spam can be very costly to e-mail recipients; it reduces their productivity by wasting their time and causing annoyance to deal with a large amount of spam. According to Ferris Research, if an employee got five e-mails per day and consumes 30 seconds on each, then he/she will waste 15 hours a year on them. Multiplying this by the hourly rate of each employ in a company will give the cost of spam to this company. In addition, spam consumes the network bandwidth and storage space and can slow down email servers. Spam software can also be used to distribute harmful content such as viruses, Trojan
horses, worms and other malicious codes. It can be a means for phishing attacks as well.

As a result, spam has become an area of growing concern attracting the attention of many security researchers and practitioners. In addition to regulations and legislations, various anti-spam technical solutions have been proposed and deployed to combat this problem. Front-end filtering was the most common and easier way to reject or quarantine spam messages as early as possible at the receiving server. However, most of the early anti-spam tools were static; for example using a blacklist of known spammers, a white list of good sources, or a fixed set of keywords to identify spam messages. Although these list-based methods can substantially reduce the risk provided that lists are updated periodically, they fail to scale and to adapt to spammers’ tactics. They can be defeated easily by changing the sender’s address each time, intentionally misspelling words, or forging the content to bypass spam filters.

The similarity of spam filters with text categorization problems and the success of machine learning techniques in solving these problems have intrigued several researchers to investigate their applicability in filtering spam. One subtle difference is that a false positive would mean that an important e-mail was identified as spam and rejected. According to a leading body in IT, inaccurate anti-spam solutions may be responsible for wasting more than five million working hours a year on checking that legitimate messages were not mistakenly quarantined. Recently, various machine-learning methods have been used to address spam filtering including support vector machines, memory-based learning, rough set, neural networks, Bayesian classifiers, sparse binary polynomial hash, etc. Among these methods, the Naïve Bayesian classifier has been widely applied as one of the most effective methods to counteract spam. A recent overview and taxonomy of current and potential solutions, both machine learning and non-machine learning, ranging from commercial implementations to ideas confined to current research, is presented in

### 1.1 What Is Spam?

- Best description: "Unsolicited Bulk E-mail"
- In human terms: bulk e-mail you didn't want, and didn't ask for

### 1.2 Effects of Spam

- Consumes network bandwidth
- Slows down E-mail Servers
- Provides medium to distribute harmful/offensive content
- Users are forced to waste time wading through their inbox.

### 1.3 Why Bother Filtering Spam?

- Seems to be about 30% to 60% of mail traffic is increasing & prediction is that by 2015 it will exceed 95%
- Legal retaliation not possible, yet
- Just plain irritating!

### 1.4 What solution is suggested?

- Machine Learning Approaches such as –
  - Bayesian Approach, Rule-Based Filtering,
  - Memory- Based…
  - The big issue with filtering:
    - Not just “how much spam does it catch?”
    - But “how many legitimate mails get caught, too?”
  - Fuzzy Similarity concentrates on False Positive Rate
  - False Positive would mean percent good caught as spam

### 3. EXISTING SYSTEM

- One subtle difference is that a false positive would be a more serious error than a false negative as a false positive would mean that an important e-mail was identified as spam and rejected.
- According to a leading body in IT, inaccurate anti-spam solutions may be responsible for wasting more than five million working hours a year on checking that legitimate messages were not mistakenly quarantined.
- Consumes network bandwidth
- Slows down E-mail Servers
- Provides medium to distribute harmful/offensive content
- Users are forced to waste time wading through their inbox.

### 3.1 Proposed System

- We explore a new approach based on Bayesian approach that can automatically classify e-mail messages as spam or legitimate.
- We study its performance for various conjunction and disjunction operators for several datasets.

### 4. Algorithms:

- **Bayesian Classification**
  1) Bayes Theorem
  2) Naïve Bayes Classifier

### 4.1 Introduction to Bayesian Classification

**What is it ?**

- Supervised Learning Method.
- Statistical method for classification
- Assumes an underlying probabilistic model
- Allows us to capture uncertainty about the model in a principled way by determining probabilities of the outcomes.
- Can solve diagnostic and predictive problems.
4.2 Why Bayesian?
- Provides practical learning algorithms
  - E.g. Naïve Bayes
- Prior knowledge and observed data can be combined
- Calculate explicit probabilities for hypothesis
- Provides a useful perspective for understanding many learning algorithms
- Provides a gold standard for evaluating other learning algorithms
- Robust to Noise in input data

4.3. Bayes Theorem and Naïve Bayes Classifier
- Named after Thomas Bayes, an English clergyman and mathematician
  - Bayesian reasoning is applied to decision making and inferential statistics that deals with probability inference
  - Using the knowledge of prior events to predict future events
  - Example: Predicting the color of marbles in a basket

4.4. The Bayes Theorem
- \[ P(h/D) = \frac{P(D|h)P(h)}{P(D)} \]
  - \( P(h) \): Prior probability of hypothesis \( h \)
  - \( P(D) \): Prior probability of training data \( D \)
  - \( P(h/D) \): Probability of \( h \) given \( D \)
  - \( P(D/h) \): Probability of \( D \) given \( h \)

4.5 DATA TABLE -1

<table>
<thead>
<tr>
<th>row</th>
<th>Age</th>
<th>Income</th>
<th>Student</th>
<th>Credit_rating</th>
<th>Buy_computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>&lt;=30</td>
<td>High</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
<tr>
<td>r2</td>
<td>&lt;=30</td>
<td>High</td>
<td>No</td>
<td>Excellent</td>
<td>No</td>
</tr>
<tr>
<td>r3</td>
<td>31...40</td>
<td>High</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r4</td>
<td>&gt;40</td>
<td>Medium</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r5</td>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r6</td>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Excellent</td>
<td>No</td>
</tr>
<tr>
<td>r7</td>
<td>31...40</td>
<td>Low</td>
<td>Yes</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r8</td>
<td>&lt;=30</td>
<td>Medium</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
<tr>
<td>r9</td>
<td>&lt;=30</td>
<td>Low</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r10</td>
<td>&gt;40</td>
<td>Medium</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r11</td>
<td>&lt;=30</td>
<td>Medium</td>
<td>Yes</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r12</td>
<td>31...40</td>
<td>Medium</td>
<td>No</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r13</td>
<td>31...40</td>
<td>High</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r14</td>
<td>&gt;40</td>
<td>Medium</td>
<td>No</td>
<td>Excellent</td>
<td>No</td>
</tr>
</tbody>
</table>

4.6 Maximum a posteriori (MAP) Hypothesis
- Generally we want the most probable hypothesis given the training data
- \( h_{MAP} = \arg \max_{h \in H} P(h|D) \)
  - \( h_{MAP} = \arg \max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \)
  - \( h_{MAP} = \arg \max_{h \in H} P(D|h)P(h) \)
  - \( h_{MAP} = \arg \max_{h \in H} P(D|h)P(h) \)

AN EXAMPLE

- D : 35 year old customer with an income of $50,000 PA
- h : Hypothesis that our customer will buy our computer
- \( P(h/D) \): Probability that customer D will buy our computer given that we know his age and income
- \( P(h) \): Probability that any customer will buy our computer regardless of age (Prior Probability)
- \( P(D/h) \): Probability that the customer is 35 yrs old and earns $50,000, given that he has bought our computer (Posterior Probability)
- \( P(D) \): Probability that a person from our set of customers is 35 yrs old and earns $50,000
4.7 Data Table

<table>
<thead>
<tr>
<th>rec</th>
<th>Age</th>
<th>Income</th>
<th>Student</th>
<th>Credit rating</th>
<th>Buys computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>&lt;=30</td>
<td>High</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
<tr>
<td>r2</td>
<td>&lt;=30</td>
<td>High</td>
<td>No</td>
<td>Excellent</td>
<td>No</td>
</tr>
<tr>
<td>r3</td>
<td>31-40</td>
<td>High</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r4</td>
<td>&gt;40</td>
<td>Medium</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r5</td>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r6</td>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Excellent</td>
<td>No</td>
</tr>
<tr>
<td>r7</td>
<td>31-40</td>
<td>Low</td>
<td>Yes</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r8</td>
<td>&lt;=30</td>
<td>Medium</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
<tr>
<td>r9</td>
<td>&lt;=30</td>
<td>Low</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r10</td>
<td>&gt;40</td>
<td>Medium</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r11</td>
<td>&lt;=30</td>
<td>Medium</td>
<td>Yes</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r12</td>
<td>31-40</td>
<td>Medium</td>
<td>No</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r13</td>
<td>31-40</td>
<td>High</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r14</td>
<td>&gt;40</td>
<td>Medium</td>
<td>No</td>
<td>Excellent</td>
<td>No</td>
</tr>
</tbody>
</table>

4.8 Maximum a posteriori (MAP) Hypothesis

- Generally we want the most probable hypothesis given the training data
- \( h_{MAP} = \arg \max P(h/D) \) (where \( h \) belongs to \( H \) and \( H \) is the hypothesis space)
- \( h_{MAP} = \arg \max \frac{P(D|h) P(h)}{P(D)} \)

**Examples:**

<table>
<thead>
<tr>
<th>S No</th>
<th>Age</th>
<th>Income</th>
<th>Student</th>
<th>Credit rating</th>
<th>Buys computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>35</td>
<td>Medium</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>High</td>
<td>No</td>
<td>Average</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>Low</td>
<td>Yes</td>
<td>Good</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>35</td>
<td>Medium</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>45</td>
<td>Low</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>35</td>
<td>High</td>
<td>No</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>35</td>
<td>Medium</td>
<td>No</td>
<td>Good</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>25</td>
<td>Low</td>
<td>No</td>
<td>Good</td>
<td>No</td>
</tr>
<tr>
<td>9</td>
<td>28</td>
<td>High</td>
<td>No</td>
<td>Average</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>35</td>
<td>Medium</td>
<td>Yes</td>
<td>Average</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Examples:**

- \( P(\text{buys computer} = \text{yes}) = \frac{5}{10} = 0.5 \)
- \( P(\text{buys computer} = \text{no}) = \frac{5}{10} = 0.5 \)
- \( P(\text{customer is 35 yrs & earns $50,000}) = \frac{4}{10} = 0.4 \)
- \( P(\text{customer is 35 yrs & earns $50,000 / buys computer} = \text{yes}) = \frac{3}{5} = 0.6 \)
- \( P(\text{customer is 35 yrs & earns $50,000 / buys computer} = \text{no}) = \frac{1}{5} = 0.2 \)

- Find whether the customer buys a computer or not?
- Customer buys a computer \( P(h1/D) = P(h1) \)
- Customer does not buy a computer \( P(h2/D) = P(h2) \)

4.9 Maximum likelihood (ML) Hypothesis

- If we assume \( P(hi) = P(hj) \)
- Where the calculated probabilities amount to the same
- Further simplification leads to
- \( h_{ML} = \arg \max P(D/hi) \) (where \( hi \) belongs to \( H \))

4.10 Naïve Bayesian Classification

- Based on the Bayesian theorem
- Particularly suited when the dimensionality of the inputs is high
- Parameter estimation for naïve Bayes models uses the method of maximum likelihood
In spite over-simplified assumptions, it often performs better in many complex real-world situations.

Advantage: Requires a small amount of training data to estimate the parameters.

### 4.11 Naive Bayes Classifier: Derivation

- **D**: Set of tuples
  - Each Tuple is an ‘n’ dimensional attribute vector
  - \( X : (x_1, x_2, x_3, \ldots, x_n) \)
- Let there be ‘m’ Classes: \( C_1, C_2, C_3 \ldots C_m \)
- NB classifier predicts \( X \) belongs to Class \( C_i \) iff
  \[
P(C_i | X) > P(C_j | X) \quad \text{for} \quad 1 \leq j \leq m, \quad j \neq i
  \]
- **Maximum Posteriori Hypothesis**
  \[
P(C_i | X) = \frac{P(X | C_i) P(C_i)}{P(X)}
  \]
  - Maximize \( P(X | C_i) P(C_i) \) as \( P(X) \) is constant.
  - With many attributes, it is computationally expensive to evaluate \( P(X/C_i) \)
  - **Naive Assumption of “class conditional independence”**
  \[
P(X/C_i) = \prod_{k=1}^{n} P(x_k/C_i)
  \]

#### 4.12 Data Table

**EXAMPLES**

<table>
<thead>
<tr>
<th>rec</th>
<th>Age</th>
<th>Income</th>
<th>Student</th>
<th>Credit rating</th>
<th>Buys computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>&lt;=30</td>
<td>High</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
<tr>
<td>r2</td>
<td>&lt;=30</td>
<td>High</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r3</td>
<td>31-40</td>
<td>High</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r4</td>
<td>&gt;40</td>
<td>Medium</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r5</td>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r6</td>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r7</td>
<td>31-40</td>
<td>Low</td>
<td>Yes</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r8</td>
<td>&lt;=30</td>
<td>Medium</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r9</td>
<td>&lt;=30</td>
<td>Low</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r10</td>
<td>&gt;40</td>
<td>Medium</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r11</td>
<td>&lt;=30</td>
<td>Medium</td>
<td>Yes</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r12</td>
<td>31-40</td>
<td>Medium</td>
<td>No</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r13</td>
<td>&gt;40</td>
<td>High</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r14</td>
<td>&gt;40</td>
<td>Medium</td>
<td>No</td>
<td>Excellent</td>
<td>No</td>
</tr>
</tbody>
</table>

For Tuple \( X \): \( \text{age= youth, income = medium, student = yes, credit rating = fair} \)

- \( P(C_1) = P(\text{buys computer = yes}) = \frac{9}{14} = 0.643 \)
- \( P(C_2) = P(\text{buys computer = no}) = \frac{5}{14} = 0.357 \)
- \( P(\text{age=youth /buys computer = yes}) = \frac{2}{9} = 0.222 \)
- \( P(\text{age=youth /buys computer = no}) = \frac{3}{5} = 0.600 \)
- \( P(\text{income=medium /buys computer = yes}) = \frac{4}{9} = 0.444 \)
- \( P(\text{income=medium /buys computer = no}) = \frac{2}{5} = 0.400 \)
- \( P(\text{student=yes /buys computer = yes}) = \frac{6}{9} = 0.667 \)
- \( P(\text{student=yes /buys computer = no}) = \frac{1}{5} = 0.200 \)
- \( P(\text{credit rating=fair /buys computer = yes}) = \frac{6}{9} = 0.667 \)
- \( P(\text{credit rating=fair /buys computer = no}) = \frac{2}{5} = 0.400 \)
- \( P(\text{X/Buys a computer = yes}) = P(\text{age=youth /buys computer = yes}) * P(\text{income=medium /buys computer = yes}) *
  \[
P(\text{student=yes /buys computer = yes}) * P(\text{credit rating=fair /buys computer = yes})
  = 0.222 * 0.444 * 0.667 * 0.667 = 0.044
  \]
- \( P(\text{X/Buys a computer = No}) = 0.600 * 0.400 * 0.200 * 0.400 = 0.019 \)
- **Find class \( C_i \) that Maximizes \( P(X/C_i) \)**
  \( \text{P(\text{student=yes /buys computer = yes})} * \text{P(credit rating=fair /buys computer = yes)} \)
  \( = 0.222 * 0.444 * 0.667 * 0.667 = 0.044 \)
- \( P(\text{X/Buys a computer = No}) = 0.600 * 0.400 * 0.200 * 0.400 = 0.019 \)

**Prediction:** Buys a computer for Tuple \( X \)

**5. Modules Details:**

There are main three modules

### 5.1 GUI Designing

- **GUI** is called Graphical User Interface.
- We call it as Designing part or simply Presentation Logic.
- Using the Swings concept in Java we allocate area for the statistical characterization for the transmitted.
- Swings in one of the Technologies where we can build the designing part.
- We will be using so many components like:
  - JPanel
  - JTextBox
5.2 Checking read-mail modules:

This module will connect with the Gmail using pop3 services. Gmail is a free, advertising-supported webmail, POP3, and IMAP service provided by Google. Gmail only supports POP3 connection with SSL, the connection is established via SSL.

The Java Mail API is a set of abstract APIs that model a mail system. The API provides a platform independent and protocol independent framework to build Java technology based email client applications. The Java Mail API provides facilities for reading and sending email. Service providers implement particular protocols.

Several service providers are included with the Java Mail API package; others are available separately. The Java Mail \textsuperscript{TM} API provides classes that model a mail system. The javax. Mail package defines classes that are common to all mail systems. The javax. mail. Internet package defines classes that are specific to mail systems based on internet standards such as MIME, SMTP, POP3, and IMAP. The Java Mail API includes the javax. mail package and sub packages.

5.3 Spam Blocking modules:

It is the processing of e-mail to organize it according to specified criteria. Most often this refers to the automatic processing of incoming messages, but the term also applies to the intervention of human intelligence in addition to anti-spam techniques, and to outgoing emails as well as those being received. Email filtering software inputs email. For its output, it might pass the message through unchanged for delivery to the user's mailbox, redirect the message for delivery elsewhere, or even throw the message away. Some mail filters are able to edit messages during processing.

Mail filters can be installed by the user, either as separate programs, or as part of their e-mail program (e-mail client). In e-mail programs, users can make personal, "manual" filters that then automatically filter mail according to the chosen criteria. Most e-mail programs now also have an automatic spam filtering function. Internet service providers can also install mail filters in their mail transfer agents as a service to all of their customers. Corporations often use them to protect their employees and their information technology assets. They make decisions based on matching a regular expression. Other times, keywords in the message body are used, or perhaps the e-mail address of the sender of the message. Some more advanced filters, particularly anti-spam filters, use statistical document classification techniques such as the naive Bayes classifier.

5.4 Login page
5.5 DATA SETS

From: Eun Jung Geumhye <info@eunjunggeumhye.com>
Date: September 16, 2008 5:12:05 PM EDT
To: TIF Alumni <tif-alumni@forums.nyu.edu>
Subject: Re: re: korea, korea tips?
Reply-To: TIF Alumni <tif-alumni@forums.nyu.edu>

Do not miss out Korea bath house when you are in Korea. I took
a massage (unisex) and my husband (purtaminum) there a couple of
years ago and they were absolutely in LOVE with spending time in
Korean bath house. It is called "jinn-til-bang". You might have to
pick around which ones are the best ones and they are all over, open
24 hours a day 7days a week. It is the best place to go especially
after you get off of a 14 hour flight.

Another place I would recommend is visiting Kyung-Joo and Andung-
Bu's home (I know no idea how close it is to Korea) if you are
interested in seeing some very old historical sites. Also, I
would recommend that you would go to the West coast of Korea. It is
not a preferable place to swim, but some place like, "Pac-lion", "Shan-
chun-poh", "Mu-ha-poh" have muddy bayes and the scenery in these places is
something else. The Yellow Sea is located between China and the
Korean Peninsula and the name comes from the sand particles that
colour the water, originating from the yellow river. I didn't
appreciate the beauty of the west coastline until I grew older.....

Have a great trip!!

---
You are currently subscribed to tif-alumni list: daniel-shiffman@forums.nyu.edu
To unsubscribe send a blank email to leave-tif-
alumni-16765897@forums.nyu.edu

dicom: Alicia L Gervasi <aig8@nps.edu>
Date: September 16, 2008 5:17:13 PM EDT
To: TIF Alumni <tif-alumni@forums.nyu.edu>
Subject: Martha Stewart's prison hopes
5.6 Spam text

6. CONCLUSION & FUTURE SCOPE:

- The MVC diagram models a wireless mobile online application.
- They together provide a graphical modeling language for the specification of an application, guide the entire modeling, implementation, and maintenance processes, and support reusable components. This framework decomposes a complex online application into modules.
- Each module is a plug-and-play unit.

REFERENCES