Naïve Bayesian Anti-spam Filtering Technique for Malay Language

Thamarai Subramaniam$^1$, Hamid A. Jalab$^2$, Alaa Y. Taqa$^3$

$^{1,2}$Computer System and Technology Department, Faulty of Computer Science and Information Technology, University Malaya
Mosul University-Computer Science Dep. Iraq

thamx@hotmail.com, hamidjalab@um.edu.my, alaa.taqa@gmail.com

Abstract: Internet technologies have accelerated the distribution of spam. Various anti-spam techniques have been development to minimize if not eliminate the spam problem. This paper introduces a Naïve Bayesian technique to combat spam problem for Malay language. An experiment is conducted using Naïve Bayesian techniques coupled with stemming process and document frequency to filter Malay language spam. The result of the experiment shows that Naïve Bayesian classifier is able to classify spam and non spam emails successfully.

Keywords: Anti-spam filters, Naïve Bayesian, text categorization, email, Machine Learning

1. Introduction

Email refers to an electronic messaging system that transmits message electronically across network which allows users to communicate with each other at a low cost while providing an efficient mail delivery system. The reliability, user-friendliness and availability of wide range of free email make it most popular and preferred communication tool. Spam, unsolicited emails are now widely viewed as a serious threat to the Internet, whereby it floods the users’ inboxes and costing businesses billions of dollars in wasted bandwidth. In 2010 the total global productivity cost of spam increased by $2 billion to $132 billion[1].

Automated anti-spam techniques are highly recommended in combating spam. As such, many researchers are devising anti-spam classifiers using various machine learning approaches. Naïve Bayesian classification technique is one of such approach that has been widely used for natural language processing. Sahami, Dumais [2] proposed the Naïve Bayesian (NB) approach to automatically classifier emails using decision theoretic framework. It is a straightforward application of the Naïve Bayes classifier to filter spam [3]. In recent years, Naïve Bayesian technique is been widely applied to filter spam [4-8].

An experiment was carried out using Naïve Bayesian classification technique to classify Malay language spam. Malay language is a branch of Austronesian family that mainly used in Malaysia, Indonesian, Brunei, and Singapore. In Malaysia it is national and official language. Many sectors and individuals, especially government officials communicate using Malay language. Currently many studies on anti-spam using machine learning techniques are focused on English-based spam. However there is a lack of studies conducted on Malay-based spam.

The objective of this paper is to develop a technique based on Naïve Bayes algorithm in filtering Malay language spam. The features computed using modified stemming process and document frequency are used to measure the effectiveness of Naïve Bayes classifier in determining the Malay spam and non-spam emails, which is the main contribution of this work. The outline of the paper is as follows: Naïve Bayesian classification technique presented in Section 2. Experimental results and discussion are shown in Sections 3 and 4, respectively. Finally the conclusion is presented in Section 5.

2. Naïve Bayesian Classification Technique

Naïve Bayes is a fundamental statistical approach based on probability proposed by Sahami, Dumais [2]. New emails are classified based on the “training set” features that already been classified into spam or legitimate email. Bayesian algorithm predicts the new email as a spam if words that appear in it have a high probability.

Naïve Bayesian algorithm as described by Lai [9], given a feature vector \( x = \{x_1, x_2, x_3, \cdots, x_n\} \) of an email, where \( x_1, x_2, x_3, \cdots, x_n \) denotes the attributes in the corpus. These attributes are a particular word that present or absent in an email belong to a category that to be predicted and by Bayesian law the probability that \( x \) belongs to \( c \) is as given in

\[
P(c \mid x) = \frac{P(c) \cdot P(x \mid c)}{\sum_{k \in \text{spam,legitimate}} P(k) \cdot P(x \mid k)}
\]

(1)

\( P(x) \) denotes the a-priori probability of a randomly picked e-mail has vector \( x \) as its representation, \( P(c) \) is also the a prior probability of class \( c \) (that is, the probability that a randomly picked e-mail is from that class), and \( P(x \mid c) \) denotes the probability of a randomly picked e-mail with class \( c \) has \( x \) as its representation.

Androustopoulos, Koutsias [10] notes that the probability \( P(x \mid c) \) is almost impossible to calculate because the fact that the number of possible vectors \( x \) is too high. In order to alleviate this problem, it is common to make the assumption...
that the components of the vector $x$ are independent in the class. Thus, $P(x|c)$ can be decomposed to

$$P(x|c) = \frac{P(c) \cdot \prod_{i=1}^{n} P(x_i|c)}{\sum_{k \in \{spam,legitimate\}} P(c = k) \cdot \prod_{i=1}^{n} P(x_i|c = k)}$$

(2)

So, using the Naïve Bayesian classifier for spam filtering can be computed as

$$C_{NB} = \arg \max_{c \in \{spam,legitimate\}} P(c) \prod_{i} P(x_i|c)$$

(3)

The simplicity and faster performance of Naïve Bayesian [4] approach has made it a very popular approach to be used in various classification fields, especially in spam filtering. The performance of Naïve Bayesian approach is reasonably dependable and excellent in different experimental setting [9]. It is simple to employ and independence allows parameters to be estimated on different data sets. Furthermore Naïve Bayesian takes short learning time [11]. However Naïve Bayesian classifier can only learn linear discriminant functions thus suboptimal for non-linearly separable concepts [12].

3. Experimental Results

An experiment was carried out to filter or classify spam and non spam emails based on probability values that are derived from training process. Classification process begins with pre-processing stage which includes tokenizing, stemming and feature selection steps.

Upon finishing initial pre-processing steps, a training process is conducted to create spam features library that consist of word probability based on Naïve Bayesian probability algorithm. Spam and non-spam emails are fed to the training process separately to produce the relevant training library.

Once this process in completed the result is combined for word probability. This work is carried out in stages as follows;

1. Collection of emails
2. Tokenization process
3. Features reduction process
4. Features selection process
5. Training process
6. Testing process

3.1 Data Collections

A total of 550 emails are used 350 emails (230 spam emails and 120 non spam emails) used for training and 200 emails (100 spam emails and 100 non spam emails) for testing. Due to lack of free benchmark dataset for Malay language, a Google’s Gmail account was setup for the purpose of collecting spam emails. Collected spam emails mainly contain information about health products, promotions, pornography, marketing and education. Non spam emails are collected from author’s personal email account that mainly contains official information from University or government sector and personal corresponding.

Only text based emails are used for this experiment, emails containing images and videos are ignored. Features from the email’s subjects and body are used for this experiment.

3.2 Tokenization

Collected emails need to be tokenized first before it can be used. The process of reducing a message to its colloquial component refers to tokenization [13]. Messages are broken up into series of tokens which then are added into a vector space to construct features vector for classification. Tokenization process will extract all the words from the message without the regards of its importance, thus producing a very large feature vector that may overwhelm the classifier. Collected emails go through a tokenization process whereby features from the subject and body of the emails are extracted into a features vector. This is accomplished by creating a tokenization program using Java programming language.

3.3 Features Reduction

Features reduction process is performed after tokenization process to reduce content obfuscations. Dimensionality reduction technique is applied to reduce features vector from unwanted words.

In order to improve the efficiency of classification, document frequency is used. Document frequency refers to the number of documents in which a feature occurs. The weight of the features is measured in term of frequency and the lower frequency i.e. less than the predetermined threshold is removed. Insignificant features that do not contribute to classification are ignored thus improving the efficiency of classifier. Mathematical form of DF as follows;

$$tf_{ij} = \frac{n_{ij}}{\sum_{k} n_{kj}}$$

(4)

Regular expression technique also applied to get rid unwanted characters such as symbols, spaces, numbers and misspelled words are identified and removed. Upper case words are converted to equivalent lowest cases words.

Further reduction is applied using stop word removal process which removes non-formative words which have high frequency that can cause complication in classification process. A stop word text file is created and checked against the features vector, if a stop word is recognized or identified than it is removed or deleted from the features vector.
Stemming, originally proposed by Porter in 1980, is a process for removing the commoner morphological and inflexional ending from words in English. A set of rules is applied iteratively to transform words to their roots or stems. Simply put, stemming is a process of reducing words to its basic form or its root word.

Malay language has its own morphology and unique structure. The prefix (eg. “meng..”, “ber..”, “peng..” ) and suffix (eg. “..an”, “..kan”, “..nya” ) of the words need to be removed in order to increase the learning speed of the classifier. However porter algorithm does produce incorrect root word for Malay word. Thus some changes were applied to the porter stemming algorithm. Figure 1 shows a segment of pseudo code for Malay word stemming process.

Removing “penge” and “peng” prefix
If wordStartWith “penge” then
Remove “penge”
Return word
Else if wordStartWith “peng” then
If wordCharAt(5) = vowel then
Remove “peng”
Replace the vowel with “k”
Return word
Else
Remove “peng”
Return word
Endif

Figure 1. Pseudo-code segment of Stemming process

3.4 Features Selection

Feature or word frequency is established by term frequency (tf); number of time the words appear in the message yield the significant of the word to the document. Each unique word in the features vector is applied with term frequency to calculate the values of total occurrences that then will be used for Naïve Bayesian probability. Table 1 shows the sample of total number of term occurrences in the feature vector.

3.5 Training Process

Once the total feature occurrences are generated, features probability is calculated for spam and non spam features. Data files containing total features occurrences for spam and non spam are then run through a probability program to tabulate the probability frequency or value. This is achieved for each of the unique words that appears in the features vector until a final value is obtained. Total number of occurrences of selected feature is divided with total of feature vector (corpus size) to obtain the probability value.

3.6 Testing Process

A new set of selected spam (100 emails) and non spam (100 emails) emails for testing undergo a similar process as of training process that is tokenization process to extract features from emails, features reduction process (stop-words, document frequency, regular expression and stemming) to remove features obfuscation, non-informative words and reduce features to root words, finally features selection process to establish term frequency.

Thereafter, the derived term frequency for each unique features are compared against the spam library for a match, if or once a match is found than that particular email is classified as a spam and otherwise if a match is absent than it is classified as non spam email.

4. Discussion

According to Guzella and Caminhas [3] classifier performances need to be evaluated on its information retrieval and decision theory. Accuracy, Precision, Recall parameters will used to calculate the effectiveness of the classifier’s performances.

Accuracy ratio is calculated using number of correctly classified spam and non spam against the total number of emails used. Precision ratio is based on the number of correctly classified spam to the number of total spam email. Recall rate refers to the rate of correctly marked spam against spam that misclassified as non spam and the number of spam identified a spam (table 2 and table 3).

Performance’s of information retrieval and decision theory are applied to check the whether the classifier has been successful. Experiment’s result shows (Table 4) that this experiment has been successful. Naïve Bayesian classifier able to classify spam and non spam emails with an accuracy of 96.0%.

Table 1. Sample of features Probability

<table>
<thead>
<tr>
<th>Probability</th>
<th>Words</th>
<th>frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0306122448</td>
<td>account</td>
<td>15</td>
</tr>
<tr>
<td>0.0816326530</td>
<td>ahlil</td>
<td>40</td>
</tr>
<tr>
<td>0.0918367346</td>
<td>akaun</td>
<td>45</td>
</tr>
<tr>
<td>0.0224489795</td>
<td>akhir</td>
<td>11</td>
</tr>
<tr>
<td>0.0816326530</td>
<td>alamat</td>
<td>40</td>
</tr>
<tr>
<td>0.0285714285</td>
<td>american</td>
<td>14</td>
</tr>
<tr>
<td>0.0551020408</td>
<td>amerika</td>
<td>27</td>
</tr>
<tr>
<td>0.0224489795</td>
<td>anak</td>
<td>11</td>
</tr>
<tr>
<td>0.0285714285</td>
<td>antarabangsa</td>
<td>14</td>
</tr>
<tr>
<td>0.0204081632</td>
<td>arah</td>
<td>10</td>
</tr>
<tr>
<td>0.0285714285</td>
<td>aset</td>
<td>14</td>
</tr>
<tr>
<td>0.0204081632</td>
<td>asli</td>
<td>10</td>
</tr>
<tr>
<td>0.0489795918</td>
<td>automatik</td>
<td>24</td>
</tr>
<tr>
<td>0.0326530612</td>
<td>awal</td>
<td>16</td>
</tr>
<tr>
<td>0.0204081632</td>
<td>baca</td>
<td>10</td>
</tr>
<tr>
<td>0.1122448979</td>
<td>baik</td>
<td>55</td>
</tr>
</tbody>
</table>

Table 2. Performance measurement
Filtering Malay language spam shows promising results and successful in which sees 96% accuracy (Figure 2). Further refinement on the stemming and feature selection process needed to improve accuracy rate. In order to avoid over-fitting and biasness training corpus need to be increased. Future improvement will concentrate on further reducing false positive by increasing training corpus and analyze the effects of stemming in classifying Malay language spam.

5. Conclusion

Spam is increasingly becoming a major headache for Internet users. Various anti-spam programs have been deployed and machine learning approaches are the most popular techniques to filter spam. This paper explores the use of Naive Bayesian technique to combat spam problem for Malay language. An experiment was conducted using Naive Bayesian technique in filtering Malay language spam shows promising results and successful in which sees 96% accuracy (Figure 2). Further refinement on the stemming and feature selection process needed to improve accuracy rate. In order to avoid over-fitting and biasness training corpus need to be increased. Future improvement will concentrate on further reducing false positive by increasing training corpus and analyze the effects of stemming in classifying Malay language spam.

<table>
<thead>
<tr>
<th>Measurements</th>
<th>Formulae</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy(A)</td>
<td>( \frac{TP + TN}{TP + TN + FP + FN} \times 100% )</td>
</tr>
<tr>
<td>Precision (p)</td>
<td>( \frac{TP}{TP + FP} \times 100% )</td>
</tr>
<tr>
<td>Recall(r)</td>
<td>( \frac{TP}{TP + FN} \times 100% )</td>
</tr>
</tbody>
</table>

Table 3. Meaning of keys used

<table>
<thead>
<tr>
<th>Keys</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP (True Positive)</td>
<td>Spam emails that correctly predicted as Spam email</td>
</tr>
<tr>
<td>TN (True Negative)</td>
<td>Non Spam emails that correctly predicted as Legitimate email (Non Spam)</td>
</tr>
<tr>
<td>FP (False Positive)</td>
<td>Non Spam emails that misclassified as Spam email</td>
</tr>
<tr>
<td>FN (False Negative)</td>
<td>Spam emails that misclassified as Legitimate email (Non Spam)</td>
</tr>
</tbody>
</table>

Out of 200 emails used (100 spam and 200 non spam emails) 97 emails and 95 emails are respectively classified correctly as spam (True positive) and non spam (True Negative). However, false positive of 2.5% and false negative 1.5% have caused the accuracy ratio achieved to be below 100% with an error rate at 4%. Precision is at 95.1% while the recall rate is 97.0%. Figure 2 shows the performance measurement result.

Table 4. Classification result using Confusion Matrix

<table>
<thead>
<tr>
<th>Actual</th>
<th>Predicted</th>
<th>Spam</th>
<th>Non Spam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spam</td>
<td>97</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Non Spam</td>
<td>5</td>
<td>95</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 2. Performances measurement results

False positive and false negative are vital in determining the effectiveness of classification. False positives produced by the NB classifier are still considered high. This may result in the need to frequently examine deleted emails which also diminish productivity.

False positive may be contributed by over-stemming, under-stemming or elimination of significant features from the feature vector during features reduction and selection process. Furthermore, more corpus for training may avoid or reduce over-fitting.

Reference


