Bayes Classifiers for Web Content Filtering

EDGAR ERLO E. FURATERO AND CHRISTIAN JOIE B. HARDER, Ateneo de Davao University

The research is about developing a web filter using text classification that recognizes the content of the webpage and checks whether the webpage contains pornographic content based on texts that are found within the webpage. The research uses Bayes Classifiers since it gives great potential in Web-Oriented documents. The research specifically shows the similarities and differences between the Naïve Bayes, Naïve Bayes Multinomial and Tree Augmented Naïve Bayes Classifier. This research is aimed for people who are interested in creating a more content-based filtering tool.

Categories and Subject Descriptors: **H.5.2 [Information Interfaces and Presentation]**: User Interfaces—*Evaluation/methodology*; **1.5.1 [Pattern Recognition]**: Models—*CHAID*

General Terms: Web Filter, Text Classification, Naïve Bayes, Naïve Bayes Multinomial, Tree Augmented Naïve Bayes

Additional Key Words and Phrases: Web mining, Pornographic filter, V-fold cross validation

1. INTRODUCTION

1.1 Background of the Study

With the growth of technology affecting the way many people live, so has the internet's influence. Even children can have access to the internet in this day and age simply by clicking a button and opening a browser. Opening browsers allows the vast knowledge that is spread upon the world, via the internet, to be within reach by the user by means of search engines and the like. However, there are some instances wherein unnecessary content are found when surfing the internet, one of which is pornography.

Pornography are among the top 20 search terms queried at the 10 leading Internet portals and search engines, making it the most prolific and harmful web content (Lee & Hui 2002). Pornography may be discovered by the user by accident when searching. But given the security tools that are available today, one could easily apply these tools to block pornographic sites. These tools are known as web-filtering tools that classify what type of webpage the user is currently viewing then gives an appropriate action based on that classification; for example, if a user goes to a porn site, whether unknowingly or not, the tool automatically recognizes what type of site it is then blocks the user access to the site.

Although there are web-filtering tools, most web-filtering tools today block websites by using keywords, URL database or content-filter. Using keywords was a great way of blocking sexual content before but since the development of computers and information has evolved, so have the words that are being marked as keywords making this method inefficient since it blocks harmless websites that contain the keyword. URL database is a great web-filtering tool but the problem with this tool is that the tool needs to register the URL before taking actions, meaning if the URL is not registered, the tool won't take action and will consider the URL as a normal webpage. Content-filters are filtering tools that check the actual content of the webpage and will take action based on the content found within the webpage. Many blockers still don't use content-filtering, especially on free web filtering applications (http://internet-filter-review.toptenreviews.com/).

Our proposed system should classify webpage through the use of attributes that are associated to pornographic and non-pornographic content but in addition, we will make the computer understand the context of the webpage by the use of semantics and check how frequent the keywords have been used. This will allow the computer to better identify and block pornographic web pages since the computer knows for what purpose these attributes are used for or what the webpage contains.

We will be using three different Bayesian Network Classifier for the system. We will first use all three classifiers and then compare them with each other to see which the best is.

1.2 Problem Statement

This study investigates the contents found within web pages and the effectiveness of using Bayes Classifiers to the solution of filtering of web pages.

Specifically, the study aims to answer the following questions:

- How does the standard web filter work? What are the parameters to consider in getting the content of the web site?
- How to avoid and reduce the wrong classification and wrong assumptions?
- What would be the best Bayes Classifier?
- What are the advantages and disadvantages of using Bayes classifiers?

1.3 Research Objectives

The study has the following general objective:

- To be able to classify pornographic web sites using Bayes classifiers.

The study has the following specific objectives:

- To identify the factors to be included in filtering web sites.
- To identify different methods in getting better probability or in training each Bayes classifier.
- To compare and contrast the three Bayes classifiers used in the study.
- To determine the potentials and areas to improve between the three Bayes classifiers that will be used on this application.

1.4 Significance of the study

The output of this study will commonly be used within family homes or public cafes since it allows the children to surf the internet without being exposed to pornographic content or preventing them from seeing these contents from another computer. This gives web browsing more audience control.

The output of the study is mostly used as a tool to improve classification of misclassified web pages. It may also be used as a prototype model for new web-filtering tools or as a reference for comparison between other web-classifiers or filter tools.

1.5 Scope and Limitations

The research covers the classification of the webpage, if the web site is pornographic. The research is also more focused to using the three Bayes Classifiers: Naïve Bayes, Selective Naïve Bayes & Naïve Bayes Multinomial; while at the same time improving them.

The research is only limited to English texts found within a web page and doesn't include image, advertisements add-on and the like. Our research also doesn't include encrypted texts found within web pages and the comments in the html. The research is also limited to only identifying web pages which do not include other methods of filtering such as black box, meaning URL blocking and PICS **(**Platform for Internet Content Selection) identification are excluded.

2. REVIEW OF RELATED WORKS

2.1 Web Filtering

A web filter is a system that can screen incoming webpage to determine whether the page should be or should not be displayed to the user. Web filter usually blocks web pages that have inappropriate content for a particular user, contents such as, pornographies, games, violence, spywares and other distracting contents. There are three commonly used methods in creating a web filter system: Blacklist, Keyword blocker and True web content filter.

Most content blockers or filters run on a passive manner. Usually programs like this simply block the user from entering the website if the URL is pre-defined in the database (Blacklist Method). Although there are some instances where in the website provides a rating of a specific content for the software to block such as Platform for Internet Content Selection (PICS) or in an integrated websites like video-sharing and search engine. But still, the most important problem with such software is that it is only efficient when the web site has already been scanned by the maintainers and registered it in the database and the fact that thousands of pages are added in the internet every now and then (John Charles Way, 2007).

After the blacklist method, a new approach have emerge which they called a "keyword blocker". This kind of program enables the user to block a page based on the words that the user registered on the software's database. This method worked fine in the early days of the internet, but during the modern days it is evaluated as too strict and indiscriminate a tool given the considerable range of web content available; Blocking the word "breast" and the filter would also block websites containing breast cancer, breast stroke for swimming, chicken recipe sites (Manfred Schulz) . Therefore a detailed analysis on content is more desired rather than a simple keyword blocking.

Content analysis is referred as "True" web content filter which is generally based on machine learning methods. It uses algorithms or weightings on representative features that software could find in a web page for it to be categorize. These features could be keywords, hyperlinks, images (Manfred Schulz). Web filters that use this method typically need to collect a large amount of data set for training, web pages collected from different categories. After the learning, the classifier should able to judge the web content according to the features. Most features are text-based in practice, because classifying text is more efficient than image analysis and can classify content in different categories other than pornography (Lin & Liu, 2008). Although, it doesn't mean that it is problematic to categorize pornography just by text.

2.2 Text Classification

Given that the features from web pages are by large text-based, text classification is essential in developing a true web content filter. The study of Yang, Y. and Liu, X. (1999), compared and evaluated the different text classification algorithms, namely Support Vector Machines (SVM), Neural Network (NN) and Naïve Bayes (BN). All the algorithms have reach 80% and above in terms of accuracy, using by harmonic average of recall and precision. The study also showed that the SVM have significant advantage when the training data is low as 10.

2.2.1. Support Vector Machine

A Support vector machine (SVM) uses a process to find a decision surface which can divide data points into classes in a multidimensional space. In its straightforward manner, training documents are represented as vectors. SVM is effective and capable of handling large number of training documents (Du, Safavi-Naini & Susilo, 2003)

The study of Hassan, S. et al. (2012) concentrated on improving text classification by adding different knowledge in the documents that is based from knowledge repositories like Word Net, Wikipedia and Wikitology. They compared Support Vector Machine (SVM) and Naïve Bayes (NB) for enriching the document from extracting knowledge from Wikitology. NB showed an improvement of almost 4 times better than the SVM. They concluded that Naïve Bayes classifier is better when external enriching is used from any external knowledge source.

2.2.2. Neural Networks

One of the methods that are widely used in artificial intelligence is Neural Networks (NN). A Network Neural network is identical on how the human brain works that is composed of many interconnected neurons artificial neurons. This method is known for its adaptability because the system can be train to change its internal states that will be reflected in the documents and categories. Although there is no question in the effectiveness of the NN, it is difficult to comprehend the decision of it because it is Black-box by nature (Lin & Liu, 2008).

Lee, P. Y. and Hui, S. C. (2008) used Neural Network (NN) in classifying non-pornographic and pornographic web pages but the output is defined in three categories which include the uncertain. They classify the page based on clustered text contents from diverse location of the webpage. Then the system separates the clustered text into four different types for weighing purposes: First, the web page title, second, the displayed contents, which includes the warning message and other viewable contents in the page. Third, the web page's own URL and other URL's that is embedded on the web site, lastly, the image tooltip, the text which usually occurs when you mouse over a picture, also other graphical text that can be found on the page.

Of course there are already studies on developing a true web content filter. But even though these filters are only classified based only from the text features of the web page, there are still problems that is crucial and tentative on these manner. Lee, P. Y and Hui, S. H. (2002) pointed out two main problems in classifying pornographic and non pornographic websites, that there were unclassified web pages due to insufficient amount of words that containing the website and misclassified web page due to word occurrence that is related to pornography.

2.2.3. Naïve Bayes

Naïve Bayesian (NB) classifiers are widely use because of its simplicity and computational efficiency. NB uses a probabilistic model wherein the probabilities or frequencies of words in a document that belongs to each category are estimated. Then these probabilities are used to estimate the most likely category of a certain test document (Lin & Liu, 2008).

The study of Patil, A. S., and Pawar, B. V. (2012) describe the implementation of NB to categorize different websites based on its home page. The websites were categorized by Hotels, Sports, Academic Institution, etc. with a total of ten categories. The data set was composed of extracted text particularly from HREF (hyperlink) label, title, META description and META keyword and all the body text contained on a home page. In process of getting the frequency of each term of the home page, the study used an interesting technique wherein they regulate such repeating keywords to reduce the impact of site promotion practice. Also, the study emphasize that the accuracy of the NB classifier is proportional to the number of training documents.

Another study on text classification using Naïve Bayes was conducted by Soukhikh, E. (2007). The study aims to classify discussions on different mobile companies such as Nokia, Ericson, Motorola, etc. The system established to classify the different mobile companies by using the forums of each company as the data set. The study also highlight that NB classifier is good in determining relatively close languages such as Danish, Swedish, Norwegian, etc. But it showed poor results in recognizing negative and positive statements based on emotions.

Xhemali, D., Hinde, C., & Stone, R. (2009) was also a study about classifying web sites into two different categories, but the focused of the study was to compare three different text classifiers namely, the Naïve Bayes, Decision Trees and Neural Networks. However the team had difficulties in creating the NN since the vocabulary is very large that it requires more memory to establish the model. Then the team proceeds with the NB, they introduced a method in getting a gradual probability of a data with little evidence which gives an improvement on the traditional Naïve Bayes and a slight edge over the Decision Trees.

In the paper of Schneider, K. (2005), he discussed about the weaknesses of the Naïve Bayes classifier, one of these is that the classifier is not modeled well, resulting to wrong assumptions or errors. The study showed some modifications to improve the classifier, that is to remove duplicate words in a document to account for business phenomena in text; to use uniform priors to avoid problems with skewed class distributions when the documents are very short.

2.3. Theoretical Framework

The following framework is based on the paper of Patil, A. S., and Pawar, B. V. (2012). Wherein, Naïve Bayesian Algorithm was use given with prior knowledge to classify different web pages into different categories.

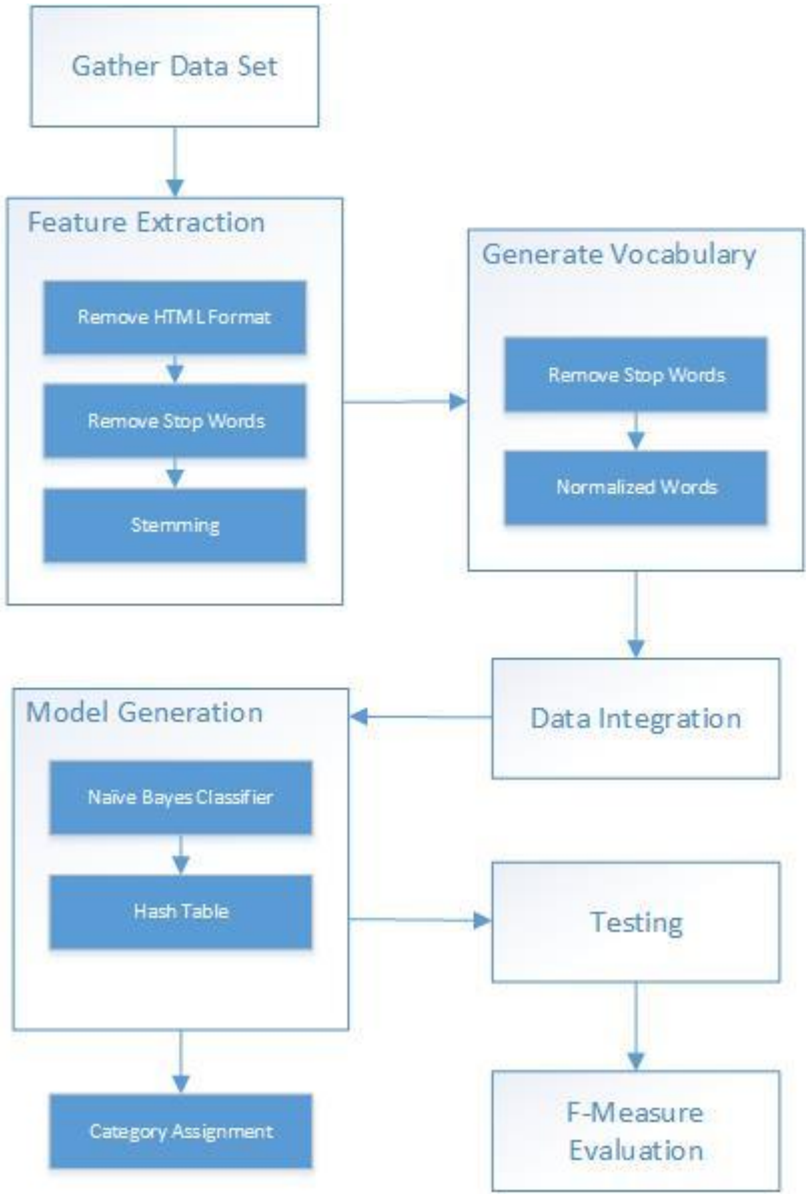


Figure 2.4 A

The first process is the gathering of different web pages from various search engines. The popular search engines like Google, Bing, Altavista etc., were submitted keyword based queries and then the results obtained were examined. Next is the feature extraction, the extraction of data or words from the web page. Then the cleaning of html formats, stemming and removing of stop words for simplicity. The next phase is the vocabulary generation; This phase involves the pre-processing of the extracted words, first, by removing additional stop words, words that are very common in websites. Then words are normalized to reduce the words that are repeating. Thus, in this phase, only the relevant words are recorded in the database and the very common and very rare words are excluded. The next process is the integration of data, wherein the development of the model is involved. The model is trained with the chosen algorithm, which in this case the Naïve Bayes. All documents that belonged to respective categories were parsed and a hash table was prepared for each category. The values of the hash table were calculated based on the frequency in all documents belonging to that category. Then the model was tested and evaluated using the F-Measure.

3. RESEARCH DESIGN AND METHODOLOGY

3.1 Conceptual Framework

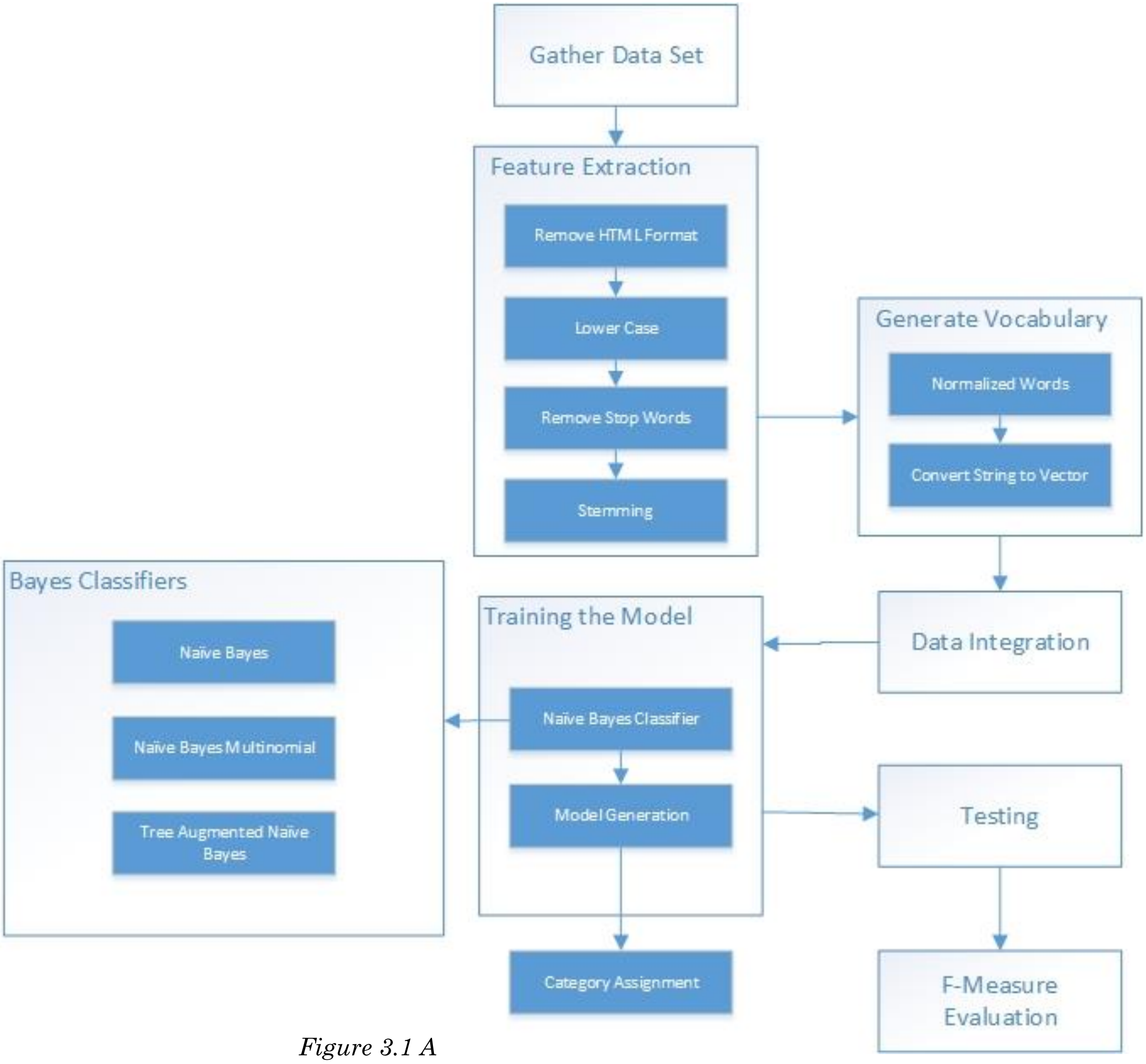


Figure 3.1 A

3.2. Methodology

These are the following phases to be followed in using Bayes Classifiers as an approach to Web Content filtering:

1) Data Gathering
2) Feature Extraction
    a. Remove Html Format
    b. Lower Case
    c. Remove Unnecessary Words
3) Generate Vocabulary
    a. Remove Stop Words
    b. Normalize Words
    c. Convert String to Vector
4) Data Integration
5) Training the Model
    a. Bayes Classifiers
        i. Naïve Bayes
        ii. Naïve Bayes Multinomial
        iii. Tree Augmented Naïve Bayes
    b. Model Generation
6) Testing
7) F-Measure Evaluation

3.3 Data Gathering

Data gathering should be done by collecting pornographic and non-pornographic web pages from popular search engines such as Yahoo, Google, Bing, etc. Also search engines with filtering options would be prioritize in collecting the data set. We would no longer check the Platform for Internet Content Selection (PICS) of each web site for additional verification since the study of Lee, P. Y and Hui, S. H. (2002) showed that 86% of the pornographic website they gathered did not use any PICS support.

3.4 Feature Extraction

This phase involves the extraction of words from the web pages. The words or the variables to be used are combined based from study of Patil, A. S., and Pawar, B. V. (2012) and Lee, P. Y. and Hui, S. C. (2008).

- Web page title
- Meta description
- Meta keyword
- HREF (hyperlink)
- Image tooltip
- Other displayed contents (paragraph, headers, etc.)

3.4.1 Remove Html Format

We will use the Jericho HTML Parser in extracting the words from the web page and also helps us to remove the unnecessary HTML formats. The Jericho HTML Parser is a powerful open source java library allowing analysis and manipulation of parts of an HTML document.

3.4.2. Lower Case

All the words are converted to lower case for a lower amount of data and to reduce redundant duplications.

3.4.3. Remove Unnecessary Words

Each words extracted from the html would be scanned and checked if the words has numeric values or unwanted symbols that makes it not a word.

3.5 Generate Vocabulary

From this phase, the Weka tool will be used. The gathered html filed would be converted to a text file then to an .arff file for the Weka tool to understand. We would also use the pre-processing features of Weka to have a more unambiguous data. After this process we would get the most relevant data and will be consider as a vocabulary of our data.

3.5.1. Remove Stop Words

The stop words will be removed based on the English stop words provided by Weka.

3.5.2. Normalized Words

We would normalize our words by limiting the amount of words to be extracted from each category.

3.5.3. String to Word Vector

After some pre-processing of words in the data, each word or attributes will be converted to a vector so that the classifiers would be applied in the data.

3.6 Data Integration

The selected words in the previous phase will be saved and will be used for training and creating the chosen classifiers for this study. The same process would be done in our chosen test set for avoiding incompatibility.

3.7 Training the Model

This section involves the creating of the model for classifying the content of the web page. The model would be trained using the desired text classification approach.

In here we would train three different Baye's Classifier. We would train the classifiers with different data numbers and changes for a possible increase in accuracy.

3.7.1 Baye's Classifiers

The approach that will be using in this study is three Bayes classifiers. Of course it uses Bayes' Theorem, a formula that calculates a probability of a word that is present in a given document. These are the specific Bayes classifier to be used in this study:

i. Naïve Bayes
ii. Naïve Bayes Multinomial
iii. Tree Augmented Naïve Bayes

3.8 Testing

Testing is done with a different set of web site documents (html for this case). The documents to be tested should not be included in the training data set.

3.9 Evaluation

We will be using the F-measure to evaluate each classifier for its efficiency and accuracy.

4 THEORETICAL BACKGROUND

4.1 Bayes Theorem

Bayes rule is a technique to estimate the likelihood of a property given the set of data as evidence or input.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{\sum_i P(B|A_i) \cdot P(A_i)}$$

In this formula A is a hypothesis and B is an observable event.
P(*A* | *B*) is the posterior probability, and P(*A*) is prior probability associated with hypothesis A.
P (*Bi*) is the probability of the occurrence of data value A, and the P(*B* | *Ai*) is the conditional probability that, given a hypothesis, a tuple satisfies it.

The other way to write the Bayes rule is:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

The power of Bayes' rule is that in many situations where we want to compute P(*A* | *B*) it turns out that it is difficult to do so directly, yet we might have direct information about P(*B* | *A*). Bayes' rule enables us to compute P(*A* | *B*) in terms of P(*B* | *A*).

4.2 Naïve Bayes

When Naïve Bayes is been used for text classification, the question will be “What is the probability that a given document *D* belongs to a given class *C*? “ Using Baye’s Rule, the formula will be:

$$p(Class|Document) = \frac{p(Class)p(Document|Class)}{p(Document)}$$

P(Document) is a constant divider common to every calculation, and can be disregarded. In testing just the words are taken into consideration (not white spaces, for example). Then the probability that document belongs to a specific class is a product of the conditional probabilities for each attribute value. In this case such attribute values will be words.

$$p(Class|Document) = p(Class)\prod_i p(Word_i|Class)$$

During the training, the application finds words that are presented more often and creates "vocabulary" for each category. When finding probability of a test document, only those words that presented in the vocabulary will be taken into consideration.

For each word appearing in the vocabulary, the conditional probability is estimated by the following formula, where *cw* is the number of times the word occurs in the category, *ccat* the number of words in the category, and *cvoc* is the size of the vocabulary.

$$\frac{1+c_w}{c_cat+c_voc}$$

4.3 Naïve Bayes Multinomial

```
-From training corpus, extract Vocabulary
-Calculate $P(c_j)$ terms
    -For each $c_j$ in $C$ do
      $docs_j \leftarrow$ all docs with class $=c_j$
      $P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$
-Calculate $P(w_k \mid c_j)$ terms
    -$Text_j$ -> single doc containing all $docs_j$
    -For each word $w_k$ in Vocabulary
      $P(w_k|c_j) \leftarrow \frac{n_k+\alpha}{n+\alpha|Vocabulary|}$
```

4.3.1 Maximum likelihood estimates:

Prior probability of class j is the count of times that the class is class j over all documents

$$\hat{P}(c_j) = \frac{N(C = c_j)}{N}$$

The likelihood of a particular word/feature is given a class. Out of all documents with a particular class, how often was the particular feature occurring with that class.

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j)}{N(C = c_j)}$$

4.3.1 Parameter estimation:

Estimates the likelihood of a particular word given a particular class by looking at the fraction of times the word occurs among all the words in the document of that class.

$$\hat{P}(X_i = w | c_j)$$

4.4 Tree Augmented Naïve Bayes

-Partition the data set in n segments

-Do n times

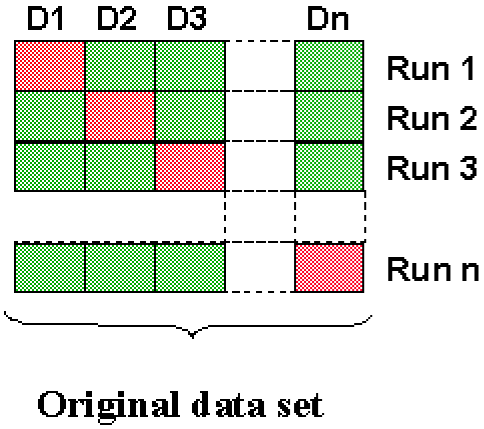
-Train the classifier with the green segments

-Test accuracy on the red segments

-Compute statistics on the n runs

-Accuracy: on test data of size m

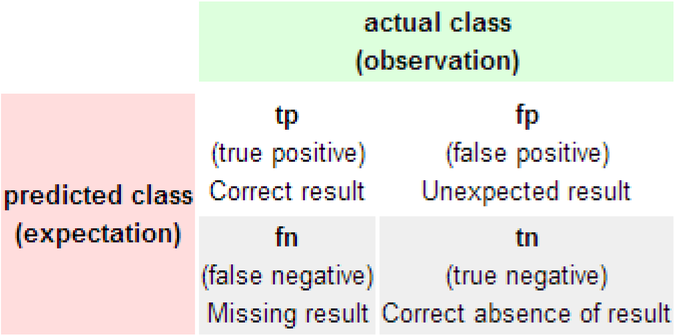
$$\mathrm{Acc} = \frac{\sum_{k=1}^{m} \lambda_k(c_i \mid l_j)}{m}$$



4.5 F-Measure

Trade-off between Precision and Recall

$$F = \frac{2 \times Precision \times Recall}{Precision + Recall}$$



	actual class **(observation)**	
predicted class **(expectation)**	**tp** (true positive) Correct result	**fp** (false positive) Unexpected result
	fn (false negative) Missing result	**tn** (true negative) Correct absence of result

4.5.1 Precision(a.k.a. True Positive Rate)

The probability that a (randomly selected) retrieved document is relevant.

$$\text{Precision} = \frac{tp}{tp + fp}$$

4.5.2 Recall(a.k.a. Positive Predictive Value)

The probability that a (randomly selected) relevant document is retrieved in a search.

$$\text{Recall} = \frac{tp}{tp + fn}$$

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

5 RESULTS AND DISCUSSIONS

5.1. Data Gathering

Data Gathering was done first, so for our dataset we manually collected web sites and save each web site as an HTML document. We collected our dataset through trusted search engines such as Google, Yahoo, Etc. We excluded web sites that present the text only on pictures and neutral websites such as web browsers and forums with no particular topic.

For our dataset, we ended up with five sub categories for the non-pornographic website and one for the pornographic category. We chose these five sub categories for the non-pornographic category since the web sites in the World Wide Web are largely belong to these classes. Also, these class or categories have words that can also be found in the Pornographic category but of course with different meaning. But another important purpose of breaking down the non-pornographic category with sub categories is to balance the documents in a category since the nature of the Baye's Theory computes for the prior knowledge thus it will have a great potential to result to a true negative if the non-pornographic will contain a larger data. Even if we balance the number of documents that can be found on both non-pornographic and pornographic category the vocabulary would still be contained by a large amount of non-pornographic words since the number of words that is associated with non-pornographic content is extremely high than the pornographic content and would still go back to the same problem.

Data Set			
Category **Sub-Category**	**Training Documents**	**Testing Documents**	**Total Documents**
Non-Pornographic			
Games	170	30	200
Health Care	173	30	203
Hotels	172	30	202
Sports	172	30	202
Tech	170	30	200
Pornographic			
Pornographic	181	32	213
Total:	**1038**	**182**	**1220**

Table 5.1 A

5.2 Feature Extraction

We created a program to extract the words from each of the html documents with the help of Jericho Html Parser library. Fortunately, Jericho Html Parser can extract html formats which were created poorly in format and in structure. So we manage to use all the documents that we collected in our data set.

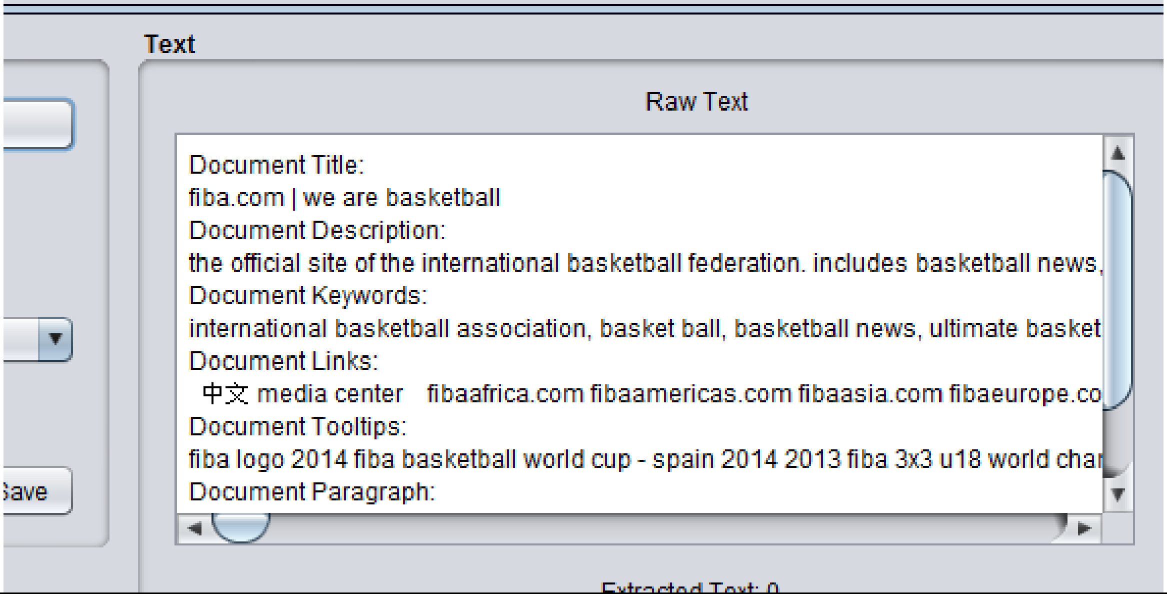
We got the necessary features from the html which was specified in the conceptual framework. Once the words were extracted, we removed the words which contain numeric values and unnecessary symbols.

Sample Java Code:

```
    import net.htmlparser.jericho.*;
    import java.util.*;
    import java.io.*;
    import java.net.*;
...

       private String extractLinks(){
          String allLabel = "";
          List<Element> linkElements=source.getAllElements(HTMLElementName.A);
          for (Element linkElement : linkElements) {
                    String href=linkElement.getAttributeValue("href");
                    //if (href==null) continue;
                    // A element can contain other tags so need to extract the text from it:
                    String label = linkElement.getContent().getTextExtractor().toString();
                    //System.out.println(label+" <"+href+'>');
                    allLabel += label + " ";
          }
          return allLabel;
   }
```

Sample Output:



```
Document Title:
fiba.com | we are basketball
Document Description:
the official site of the international basketball federation. includes basketball news,
Document Keywords:
international basketball association, basket ball, basketball news, ultimate basket
Document Links:
 中文 media center   fibaafrica.com fibaamericas.com fibaasia.com fibaeurope.co
Document Tooltips:
fiba logo 2014 fiba basketball world cup - spain 2014 2013 fiba 3x3 u18 world char
Document Paragraph:
```

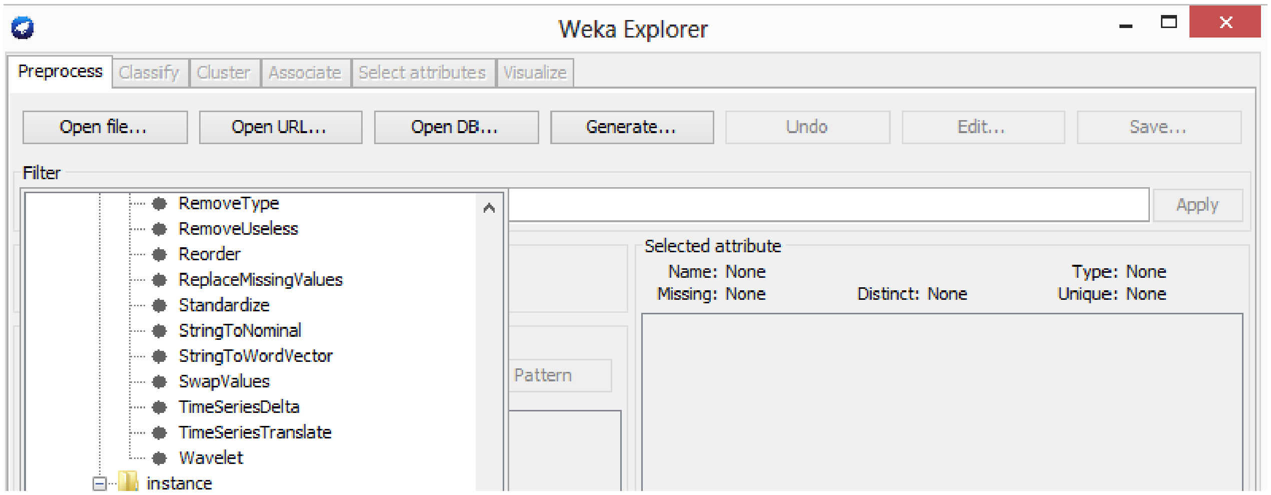
5.3 Vocabulary Generation

Weka is a collection of machine learning algorithms for data mining tasks Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

After cleaning the retrieve text from the html documents we converted it into an ".arff" using the command line in Weka so that we could feed the it in Weka, we generated our vocabulary with the help of another pre-processing method provided by said Tool. We only selected the words that occur 7 times in the whole dataset and removed the English stop words, such as that, the, is, a, etc. And only the top 2000 words that is most relevant is recorded, so that only words that occurs very rare and words that occur almost every time would be excluded in our vocabulary making the classifier more efficient.

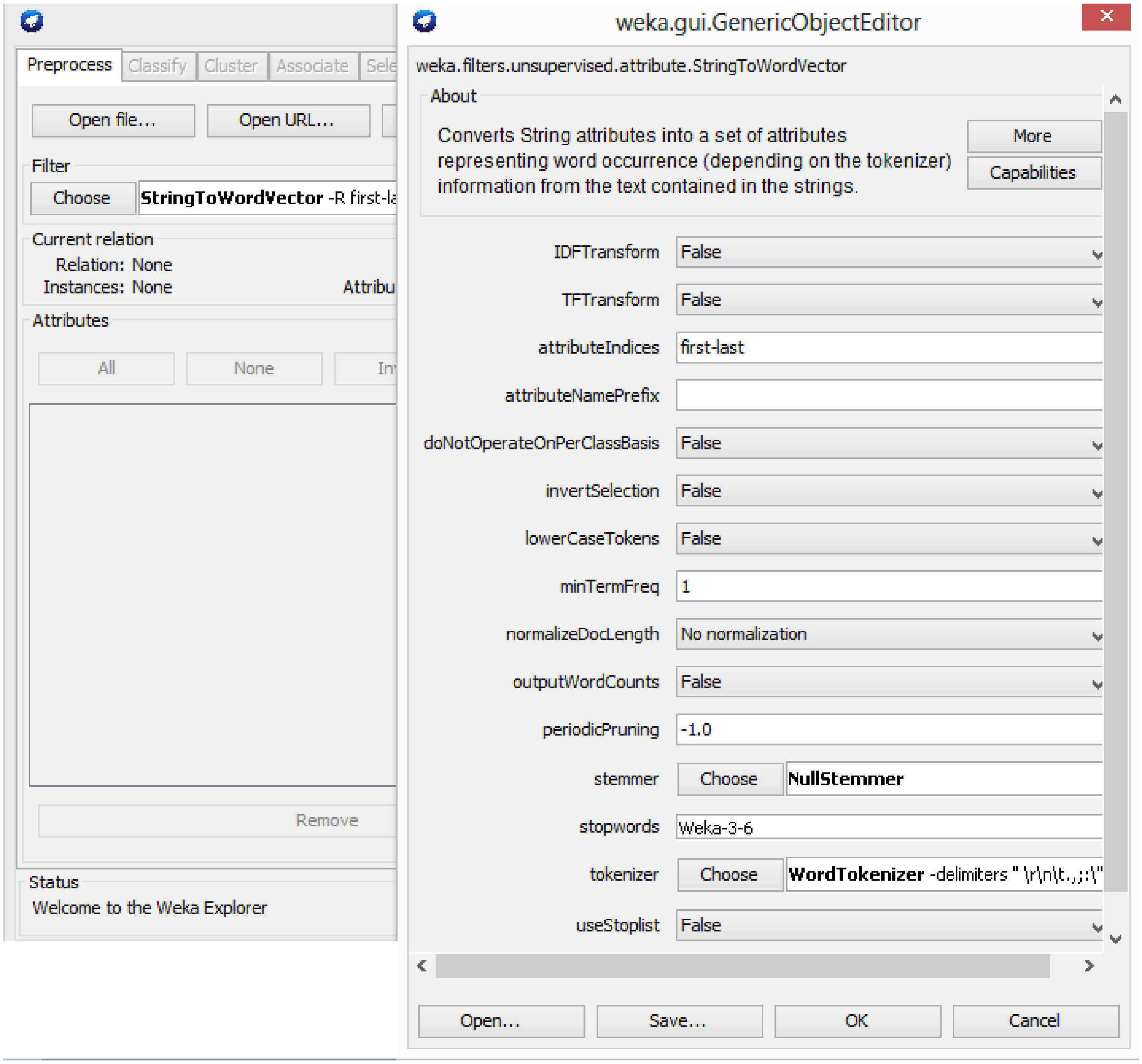
Using Weka, we also converted each String of words selected for our vocabulary to a vector by these processes:

After selecting the dataset(.arff) in the preprocess tab. Select a filter for the pre-processing method, in our case "StringToWordVector".



Name: None
Missing: None
Distinct: None
Type: None
Unique: None

Once a filter is chosen, select the filter to specify a pre-process method such as, the stop words, frequency count, etc. After, select ok, and then apply.



Filter: **StringToWordVector** -R first-la

Relation: None
Instances: None
Attribu

Welcome to the Weka Explorer

weka.filters.unsupervised.attribute.StringToWordVector

Converts String attributes into a set of attributes representing word occurrence (depending on the tokenizer) information from the text contained in the strings.

```
IDFTransform              False
TFTransform               False
attributeIndices          first-last
attributeNamePrefix
doNotOperateOnPerClassBasis  False
invertSelection           False
lowerCaseTokens           False
minTermFreq               1
normalizeDocLength        No normalization
outputWordCounts          False
periodicPruning           -1.0
stemmer                   NullStemmer
stopwords                 Weka-3-6
tokenizer                 WordTokenizer -delimiters " \r\n\t.,;:\'"
useStoplist               False
```

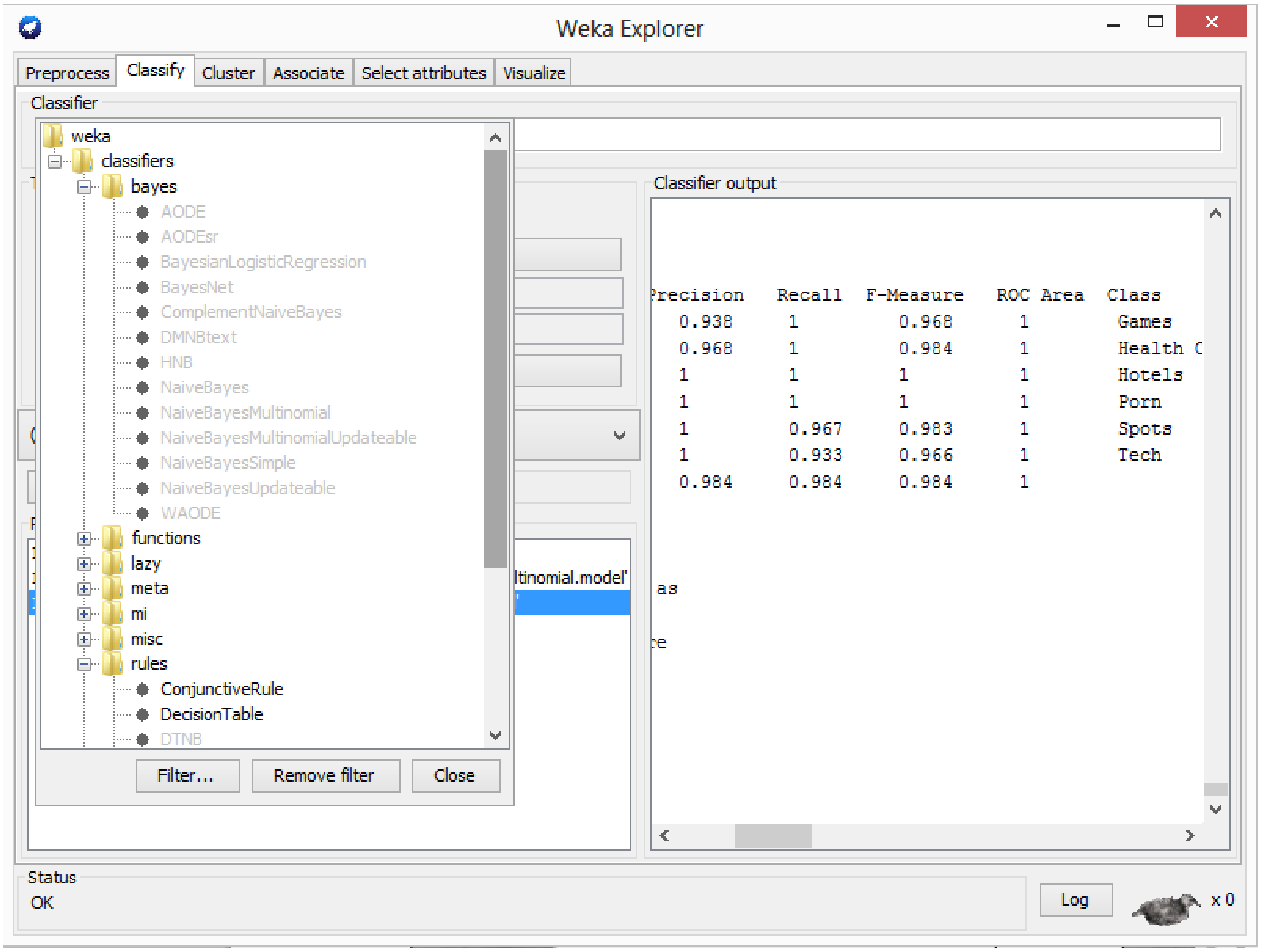
After pre-processing words that will be included in the vocabulary will be save as a word vector in an “.arff” format.

Sample of an “.arff” file

```
    @relation'D__thesisDataset_dataset200Bal_datasetTrain-weka.filters...
    @attribute @@class@@ {Games,'Health Care',Hotels,Porn,Spots,Tech}
    @attribute abdominal numeric
    @attribute abused numeric
    @attribute academic numeric
    ...
    @attribute zen numeric
    @attribute zone numeric
    ...
    @data
    {13 1,16 1,29 1,34 1,38 1,64 1,126 1,139 1,140 1,141 1,157 1,159 1,160 1,165 1,172 1,173 1,174 1,193
1,216 1,223 1,228 1,236 1,244 1,257 1,289 1,298 1,299 1,307 1,324 1,326 1,338 1,360 1,378 1,380 1,390
    ...
```

5.4 Training the Classifiers.

After selecting the dataset in Weka, you can choose a specific classifier by clicking the Classify tab. In there you can see the results, load and save a model and test a specific data.



```
Precision   Recall   F-Measure   ROC Area   Class
  0.938      1         0.968       1         Games
  0.968      1         0.984       1         Health C
  1          1         1           1         Hotels
  1          1         1           1         Porn
  1          0.967     0.983       1         Spots
  1          0.933     0.966       1         Tech
  0.984      0.984     0.984       1



 as

e
```

Before Training all the classifiers, we have done some experiment on the first classifier, the Naive Bayes. Among the three classifiers, we chose to experiment on the Naive Bayes not just because it is the fastest among the three but also because it has the core algorithm and simplest implementation from the Baye's theory. So whatever is the result of this experiment would also reflect on the other two classifiers.

Naive Bayes classifier will be train with three different arrangements of datasets but the same content of documents. For each category in the dataset, a proportion of 15% would be measured as the Testing set. The first one, all the gathered non-pornographic content would be group in one class and the pornographic will be as is (Figure 5.4.1 Dataset A). Second, all of the non-pornographic would still be in a same class but this time the instances or the number of documents in that class would be balanced on the pornographic class by, choosing 36 documents in random from each sub-category (Figure 5.4.2 Dataset B). Lastly, the non-pornographic was would be divided with different subcategories and the pornographic will remain as it is, and all of them would have an equal number of instances (Figure 5.4.3 Dataset B). Each dataset would have the same pre-processing manner to get rid of bias and a total of almost 2000 words in the vocabulary per dataset.

The result of the training would be the basis of what dataset to use on training the two remaining classifier.

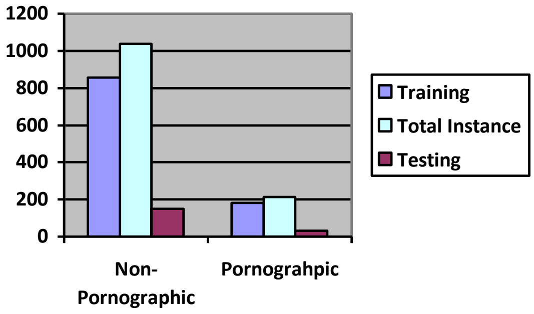


Figure 5.4.1 Dataset A

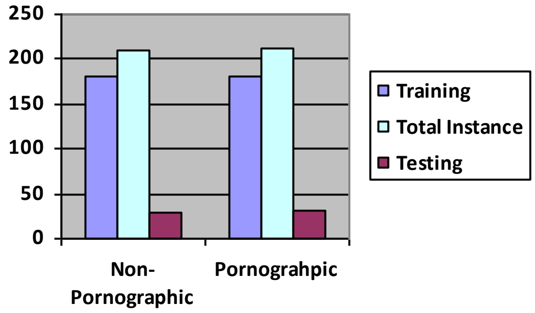


Figure 5.4.1 Dataset B

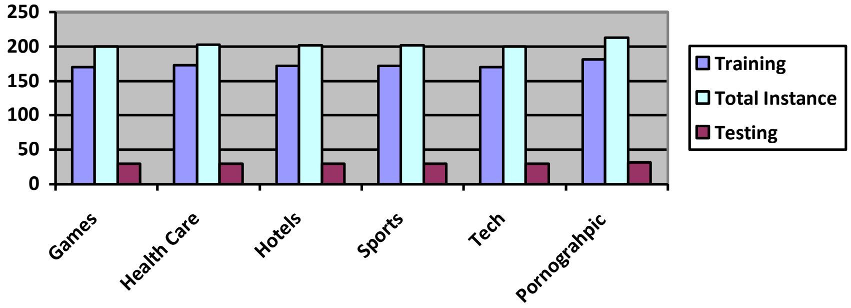


Figure 5.4.1 Dataset C

5.5 Testing and Evaluation

We will test %15 of the total instances in every category. For the Figure 5.4.3 Dataset C, a total of 182 documents will be conducted for testing. We'll be using Recall, Precision and F- Measure.

5.5.1 Testing the Experiment on the Naive Bayes

Naive Bayes checks the a word in a test document if that word is present or absent in that particular category, if present if it will have a corresponding value base on the computed posterior of that word. The category that would return the largest amount of value would be considered as the category of the test document.

With the datasets given above, section 5.4, Dataset 3 got the best accuracy in terms of classifying pornography. Although, of all the three dataset there were no instance where a non-pornographic document is classified as pornographic we would still only compare the F-Measure of the pornographic category for a more realistic comparison, since it is the constant variable in those datasets (Table 5.5.1.1).

Naive Bayes Accuracy on Pornographic Category			
Pornography on:	**Precision**	**Recall**	**F-Measure**
Dataset A	1	0.781	0.877
Dataset B	1	0.80	0.84
Dataset C	1	0.854	0.921

Table 5.5.1.1

5.5.2 Testing the three Bayes Classifier

After choosing the dataset in the experiment, Dataset C was also used to train and test the other two classifiers, namely, Naive Bayes Multinomial and Tree Augmented Naive Bayes. Unlike the traditional Naive Bayes, Naive Bayes Multinomial does not only check for the presence of the word, but also includes the frequency of that word in a category which truly adds a factor to it. While for Tree Augmented Naive Bayes, it uses nodes to find a relation of a word to another word which is also very unlikely to the traditional Naive Bayes to have independency to every word.

Naive Bayes on Dataset C			
Category	**Precision**	**Recall**	**F-Measure**
Games	0.833	1	0.909
Health Care	0.968	1	0.984
Hotels	0.968	1	0.984
Sports	1	0.967	0.983
Tech	1	0.933	0.966
Porn	1	0.844	0.915

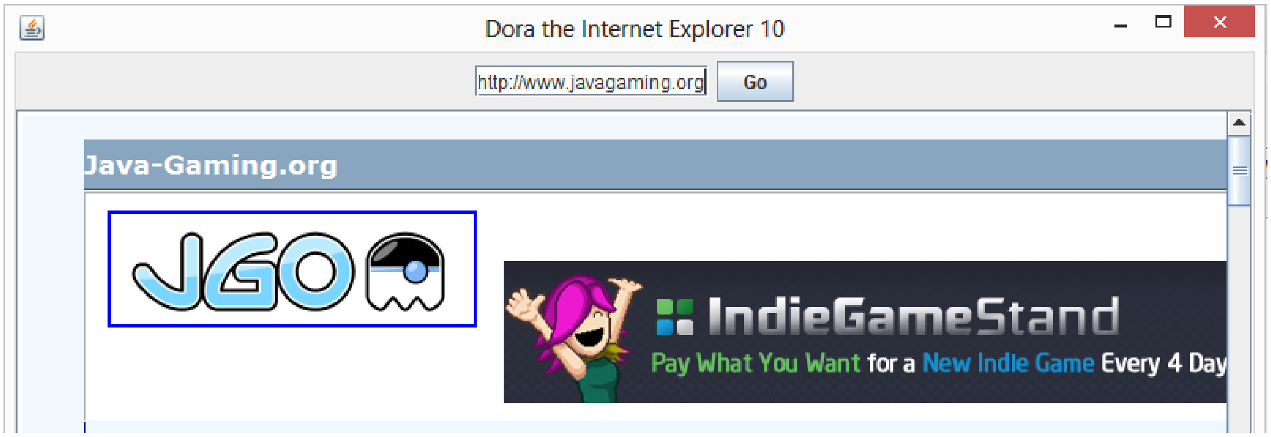
Naive Bayes Multinomial on Dataset C			
Category	**Precision**	**Recall**	**F-Measure**
Games	1	1	1

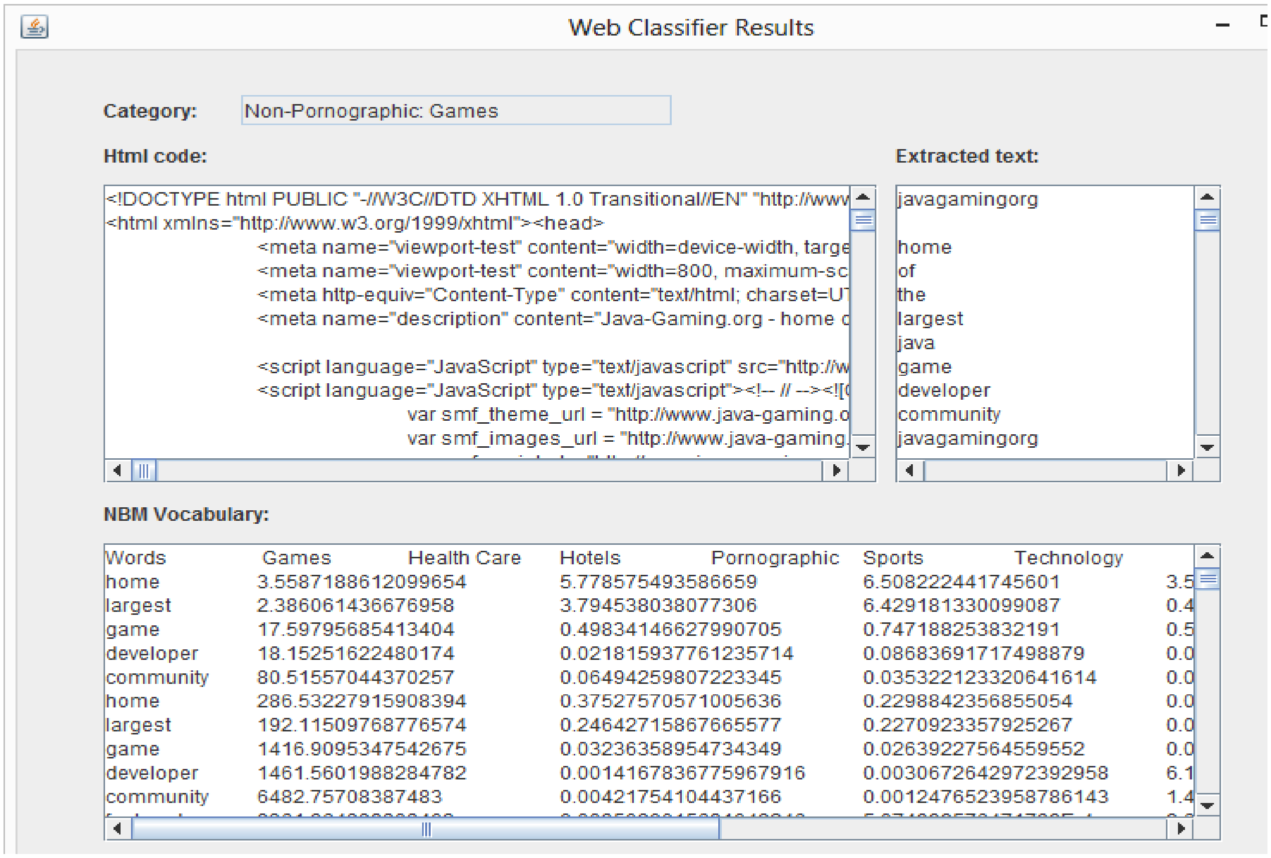
Health Care	0.968	1	0.984
Hotels	1	1	1
Sports	1	0.967	0.983
Tech	1	1	1
Porn	1	1	1

Tree Augmented Naive Bayes on Dataset C			
Category	**Precision**	**Recall**	**F-Measure**
Games	0.938	1	0.968
Health Care	0.968	1	0.984
Hotels	1	1	1
Sports	1	0.967	0.983
Tech	1	0.933	0.966
Porn	1	1	1

After the results are shown, we decided to test the Naive Bayes Multinomial classifier in our own program which is similar to a web browser. First we extracted the values from the trained model which was generated by Weka for it to be easier understood by our program and also to avoid unnecessary result. We extracted the vocabulary, the likelihood of each word and prior of each of the categories.

For our browser, we extracted the features of the website as soon as its address entered in the navigation bar. In order for the program to classify, it would check if the words from the website are found in the vocabulary and calculate its likelihood and as well as the prior. All the words that were not in the vocabulary were ignored. The highest value that was returned would determine the category.


Dora the Internet Explorer 10
http://www.javagaming.org
Go
Java-Gaming.org
JGO
IndieGameStand
Pay What You Want for a New Indie Game Every 4 Day



Category: Non-Pornographic: Games

Html code:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www
<html xmlns="http://www.w3.org/1999/xhtml"><head>
                <meta name="viewport-test" content="width=device-width, targe
                <meta name="viewport-test" content="width=800, maximum-sc
                <meta http-equiv="Content-Type" content="text/html; charset=U
                <meta name="description" content="Java-Gaming.org - home c

                <script language="JavaScript" type="text/javascript" src="http://w
                <script language="JavaScript" type="text/javascript"><!-- // --><![C
                                var smf_theme_url = "http://www.java-gaming.o
                                var smf_images_url = "http://www.java-gaming.
```

Extracted text:

```
javagamingorg

home
of
the
largest
java
game
developer
community
javagamingorg
```

NBM Vocabulary:

```
Words           Games           Health Care     Hotels          Pornographic    Sports          Technology
home            3.5587188612099654              5.778575493586659               6.508222441745601               3.5
largest         2.386061436676958               3.794538038077306               6.429181330099087               0.4
game            17.59795685413404               0.4983414662799070 5            0.747188253832191               0.5
developer       18.15251622480174               0.021815937761235714            0.08683691717498879             0.0
community       80.51557044370257               0.06494259807223345             0.035322123320641614            0.0
home            286.53227915908394              0.3752757057100563 6            0.2298842356855054              0.0
largest         192.11509768776574              0.24642715867665577             0.2270923357925267              0.0
game            1416.9095347542675              0.03236358954734349             0.02639227564559552             0.0
developer       1461.5601988284782              0.0014167836775967916           0.0030672642972392958           6.1
community       6482.75708387483                0.004217541044371 66            0.0012476523958786143           1.4
```

RESULTS AND DISCUSSIONS

6.1. Conclusion

The Bayes Classifiers were used an approach in filtering Pornographic Web sites by utilizing its features found in the Page itself. Bayes Classifiers showed great promising result but there areas to be highly considered before proceeding to model generation, it is better to balance the dataset. Not just the number of instance but also the range of relevant words in each category.

Naïve Bayes, Naïve Bayes Multinomial and Tree Augmented Naïve Bayes have reach an accuracy of %90 above but surprisingly, Naïve Bayes Multinomial have reach the highest accuracy and F-Measure in all category compared to the Naïve Bayes and a more advance classifier such as the Tree Augmented Naïve Bayes. A more advanced algorithm would not always be greater than the other. Thus, it's a good practice to study first the dataset, problem and application.

6.2. Recommendations

In using this approach as a web filter, our approach is still in need of more categories so it is recommended that in fully using this approach is to have many data.

Using this classifier for unorganized documents data is much recommended because it gathers the words even though the words are scattered.

Improvements on feature selection is still lacking because it does not register different word.

4. REFERENCES

1. 2013 best internet filter software reviews and comparisons. (2013). Retrieved from http://internet-filter-review.toptenreviews.com/

2. Definition of pornography. (n.d.). Retrieved from http://www.merriam-webster.com/dictionary/pornography

3. Du, R., Safavi-Naini, R., & Susilo, W. (2003). Web filtering using text classification. Retrieved from http://ro.uow.edu.au/cgi/viewcontent.cgi?article=1164&context=infopapers

4. Hassan, S., Rafi, M., & Shahid Shaikh, M. (2012).Comparing svm and naïve bayes classifiers for text categorization with wikitology as knowledge enrichment. (NUCES-FAST, Karachi Campus)Retrieved from http://arxiv.org/ftp/arxiv/papers/1202/1202.4063.pdf

5. John Charles Way. (2007). Meeting the challenges of web content filtering. In White Paper. Retrieved from http://dansguardian.org/downloads/content_filtering_challenges.pdf

6. Kastleman, M. (n.d.). Children as victims. Retrieved from http://www.netnanny.com/learn_center/article/144

7. Lee, P. Y., & Hui, S. C. (2002). Neural networks for web content filtering pui. Retrieved from http://classweb.gmu.edu/kersch/infs770/Semantic_Web_16_2/Neural Networks Web Content.pdf

8. Manfred Schulz. Internet Content Filtering For Administrators. netsentron. In Surray. Retrieved from http://www.netsentron.com/pdfs/ContentFilteringforAdmins.pdf

9. Naive bayes. (n.d.). Retrieved from http://docs.oracle.com/cd/B28359_01/datamine.111/b28129/algo_nb.htm

10. Patil, A. S., & Pawar, B. V. (2012). Automated classification of web sites using naive bayesian algorithm. Retrieved from http://www.iaeng.org/publication/IMECS2012/IMECS2012_pp519-523.pdf

11. Po-Ching LIN, M. L. (2008). Accelerating web content filtering by the early decision algorithm. In Retrieved from http://speed.cis.nctu.edu.tw/~ydlin/ieice.pdf

12. Schneider, K. (2005). Techniques for improving the performance of naive bayes for text classification. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.59.2085

13. Soukhikh, E. (2007). Classification of web-based discussions using naive bayes. (Thesis, Agder University College).

14. Wang, C., Lu, J., & Zhang, G. (2007). A semantic classification approach for online product reviews. (Thesis, University of Technology, Sydney)Retrieved from http://epress.lib.uts.edu.au/research/bitstream/handle/10453/2618/2005003119.pdf?sequence=1

15. Xhemali, D., Hinde, C., & Stone, R. (2009). Naïve bayes vs. decision trees vs. neural networks in the classification of training web pages. Retrieved from http://cogprints.org/6708/

16. Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. In Retrieved from http://www2.hawaii.edu/~chin/702/sigir99.pdf