ORIGINAL RESEARCH

An empirical evaluation of text classification and feature selection methods

Muazzam Ahmed Siddiqui*

Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Saudi Arabia

Received: March 20, 2016 **Accepted:** April 27, 2016 **Online Published:** June 23, 2016
DOI: 10.5430/air.v5n2p70 **URL:** http://dx.doi.org/10.5430/air.v5n2p70

ABSTRACT

An extensive empirical evaluation of classifiers and feature selection methods for text categorization is presented. More than 500 models were trained and tested using different combinations of corpora, term weighting schemes, number of features, feature selection methods and classifiers. The performance measures used were micro-averaged F measure and classifier training time. The experiments used five benchmark corpora, three term weighting schemes, three feature selection methods and four classifiers. Results indicated only slight performance improvement with all the features over only 20% features selected using Information Gain and Chi Square. More importantly, this performance improvement was not deemed statistically significant. Support Vector Machine with linear kernel reigned supreme for text categorization tasks producing highest F measures and low training times even in the presence of high class skew. We found statistically significant difference between the performance of Support Vector Machine and other classifiers on text categorization problems.

Key Words: Classifier comparison, Feature selection methods, Empirical evaluation, Text categorization, Class imbalance

1. INTRODUCTION

This paper presents an empirical evaluation of text categorization and feature selection methods applied on five benchmark corpora. While, extensive studies[1–4] in the past have been conducted, this work investigated additional measures and resources that were not covered before. These measures include training time, term weighting schemes, additional corpora, effect of number of categories, number of attributes and number of instances on performance.

Text categorization refers to the process of classifying text documents to one of the predefined classes. There are two important characteristics that distinguish text categorization or classification from other classification problems. The first and foremost is the high dimensionality of data and ensuing problems such as sparseness and noise. The high dimensionality is a direct consequence of using the vector space model that represent the text (or document) as a vector of words. It follows that the text collection, *i.e.* the corpus, is thus represented as a set of such vectors where the elements consist of all the words in the corpus, resulting in high dimensionality. The second important characteristic of the text categorization problem is the number of categories. While more than two category problems occur in other domains too, text categorization takes this as a norm.

Mathematically, the text categorization problem can be stated as:

$$P(C_i|D) = \frac{P(D|C_i)P(C_i)}{P(D)}$$

Where, $P(C_i|D)$ is the posterior probability of assigning

*Correspondence: Muazzam Ahmed Siddiqui; Email: maasiddiqui@kau.edu.sa; Address: Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Saudi Arabia.

document D to category C_i, $P(C_i)$ is the prior probability of the C_i, $P(D|C_i)$ is the class conditional probability, *i.e.* the likelihood of the finding D in C_i and $P(D)$ is called the evidence. Text categorization can also be seen as the problem of establishing decision boundaries in the high dimensional feature space. A classifier is a method that solves the $P(C_i|D)$ problem or establishes these decision boundaries. A classifier is called generative, if the class conditional probabilities in (1) are estimated explicitly from the training examples. If the estimation of class conditional probabilities is skipped and the decision boundaries are directly established, the classifier is termed as discriminative. If any assumption is made about the joint probability distribution of the features, the classifier is called parametric, otherwise it is called non-parametric. Building a parametric classifier amounts to estimating the parameters of the assumed probability distribution. A non-parametric classifier skips this step.

Dimensionality reduction refers to finding a reduced representation of the data. Because of the high dimensionality of the text categorization problem, reducing the number of dimensions to a sizable value for speed and memory consideration as well as noise reduction is important. The latter term refers to the removal of noise words, *i.e.* the words that do not help in the categorization task.

In this paper we investigated the effect of various factors on text categorization. The performance measures were the micro-averaged F measure and training time for the classifiers while the factors under investigation include the following:

(1) Term weighting schemes
(2) Number of features
(3) Feature selection method
(4) Classifier
(5) Categories
(6) Document length

We built more than 500 models using various combinations of the above-mentioned criteria with five benchmark corpora and five different classifiers. The main contribution of this work comes from the extensive empirical investigation of the effect of above-mentioned factors on the classifier performance.

The rest of the paper is organized as follows: Section 2 describes previous evaluations of classifiers and feature selection method for text corpora. Section 3 describes the corpora used in this study. Section 4 describes the classifiers, feature selection and validation methods used in this investigation. Section 5 presents the results and discussion.

2. Related work

There have been several studies in the past to compare text classification and feature selection methods. Most notable of them were carried out by References.[1–3,5] These studies investigated different aspects of text categorization such as the effect of feature selection, feature selection methods, corpus characteristics, *e.g.* number of classes, class skewness *etc*. and the type of classifier or individual classifiers. In their seminal work of feature selection in text categorization, Yang and Pedersen[1] compared five feature selection methods to determine their effectiveness on classifier performance. Experiments were conducted on two corpora using two classifiers. They reported IG and Chi to perform similarly and indicated that only top 2% of the features selected through IG produced better performance than using all the features on one of the corpus. In a separate work, Yang[3] presented an extensive comparison of 14 classifiers on two different corpora. They reexamined previously published results and presented new insight on their validity. Term weighting techniques were also addressed. Feature reduction was applied to two of the classifiers (KNN and LLSF) to make them computationally tractable. Incidentally these were the classifiers that reported the best results. Yang and Liu[2] reexamined four text categorization methods that they used earlier and added SVM to the mix. An important contributing factor was that the validation of comparison results through statistical significance tests. Only one corpus was used in the experiments, this time. SVM and KNN were reported to be the best classifiers while NB achieved the lowest performance. Feature selection for text categorization was again covered in an extensive empirical study.[4] They compared twelve feature selection methods on 229 binary text classification problems extracted from three corpora. They investigated the class skewness issue in text categorization and evaluated the feature selection methods in light of this issue. Even though pilot studies used several classifiers, results were reported for SVM only. A little known feature selection method Bi-Normal Separation (BNS) achieved the best performance in their experiment. The effect of feature selection on classifier performance in text categorization was studied by Lewis.[5] The experiments were conducted using expected mutual information as the feature selection method and a probabilistic classifier for text categorization on two corpora.

A direct comparison of results for the above mentioned studies is very difficult as different performance measures were used by each. Yang and Pedersen[1] used precision and recall, Yang also[2] used micro-averaged F measure, and micro-averaged F measure, precision, recall and also macro-averaged F measure[3] to report the results. Forman[4] used macro-averaged F measure and precision and recall.

Besides empirical studies, several surveys exist[6–9] that provide an overview and some theoretical insight into the text categorization problem.

Our work combined the different aspects of text categorization in a single experimental setup where the effects of feature selection, feature selection methods, term weighting schemes, class skew and classifiers on text categorization were investigated. Previous studies were limited to two corpora only, while we included five benchmark corpora to incorporate different corpus sizes, number of categories and category skew. In addition, we also reported training time for different corpora and different classifiers, an important aspect that was not covered in the above mentioned studies.

3. CORPORA

We used five benchmark corpora in this study. These corpora cover different number of categories, document (instances) and number of terms. It should be noted that different versions of some of these corpora exist. Table 1 describes the name and source of each of these corpora.

Table 1. Corpus version and source

No	Name	Version
1	Movies	Polarity dataset v2.0*
2	20 Newsgroups	20 Newgroup#
3	Reuters 21578	Reuters-21578 Apte-90 categories#
4	Ohsumed	All Cardiovascular Diseases#
5	IMDB movie reviews	Large Movie Review Dataset v1.0†

* https://www.cs.cornell.edu/people/pabo/movie-review-data/
# http://disi.unitn.it/moschitti/corpora.htm
† http://ai.stanford.edu/~amaas/data/sentiment/

The Reuters-21578 corpus is pre-divided into training and test corpora while the test are available as one single collection. To keep the validation method uniform across all the corpora we combined the Reuters-21578 training and testing corpora into a single corpus. The size of the 20 Newsgroups, Reuters-21578 and Ohsumed corpora are reported by Moschitti,[10] but the downloaded corpora showed slightly different statistics. Table 2 displays the statistics for each corpus. To tokenize the text, a simple non-letter based tokenizer was used. This tokenizer splits the text using the non-letter characters.

In Table 2, Types refer to the number of unique terms, *i.e.* vocabulary, while Tokens refer to the total number of terms. TTR (Type Token Ratio) describe the vocabulary diversity or richness. Length is the average number of tokens per document. Large corpora have low TTR as the tokens increase with the number of documents while the types do not increase by the same amount as they are coming from a fixed vocabulary. Figure 1 plots TTR against the number of documents and a decreasing trend can be spotted. Another factor that could affect TTR is the domain of the corpus. The lowest TTR is reported for the Ohsumed corpus that, besides being the largest corpus, belongs to a single domain of medical abstracts. Corpora spanning different domains have higher TTR. Movies and IMDB are both movie review corpora. Reuters-21578 consists of newswire documents belonging to 91 different topics. 20 Newsgroups is a collection of newsgroup documents from 20 topics. The sample containing five corpora is too small to establish a cause and effect relationship between TTR and corpus domain and was not investigated further.

Table 2. Corpus statistics

Corpus	Documents	Categories	Types	Tokens	TTR	Length
Movies	2,000	2	38,911	1,331,272	2.92%	361.2
Reuters-21578	15,437	91	44,336	2,215,495	2.00%	88.9
20 Newsgroups	20,417	20	121,603	5,294,589	2.30%	148.3
IMDB movie reviews	50,000	2	129,340	11,915,178	1.09%	128.8
Ohsumed	56,984	23	72,909	9,867,373	0.74%	108.8

The corpora in this study were carefully selected to represent a diverse collection of number of documents, terms and categories as displayed by the 3D scatter plot in Figure 2. The size of the bubble represents the number of categories. Reuters-21578 has the largest number of categories, followed by Ohsumed and 20 Newsgroups. Movies and IMDB are both binary classification problems with very different corpus sizes. Figures 3-7 display the category distribution for the Movies, IMDB, Reuters-21578, Ohsumed and 20 Newsgroup corpus respectively. Movies and IMDB have an equal and 20 Newsgroup has an almost equal class distribution. Reuters-21578 and Ohsumed have highly imbalance classes, the effect of that was obvious in the results. In addition, the latter two corpora are multi-label, *i.e.* a document can be assigned to more than one category, making categorization more difficult.

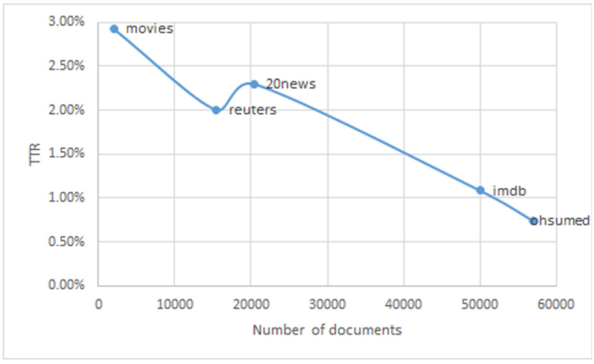

Figure 1. TTR *vs.* number of documents

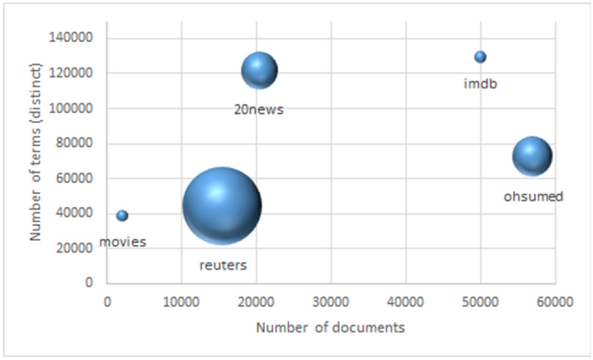

Figure 2. Number of terms vs number of documents for each corpus

Figure 3. Category distribution for the Movies corpus

Figure 4. Category distribution for the IMDB corpus

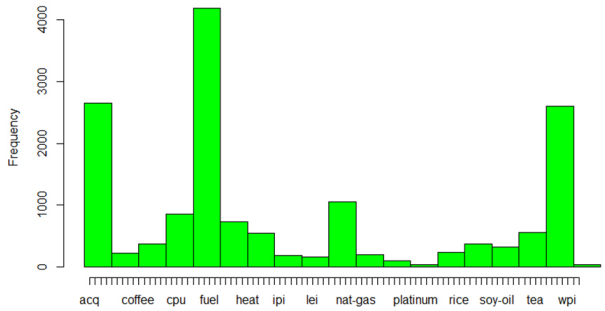

Figure 5. Category distribution for the Reuters-21578 corpus

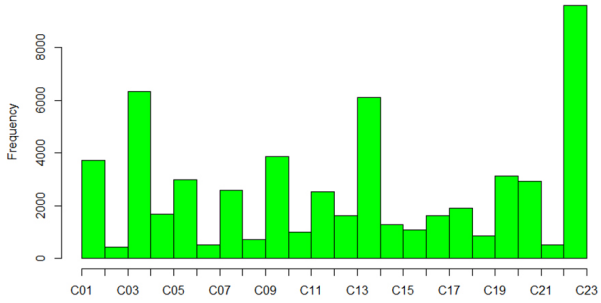

Figure 6. Category distribution for the Ohsumed corpus

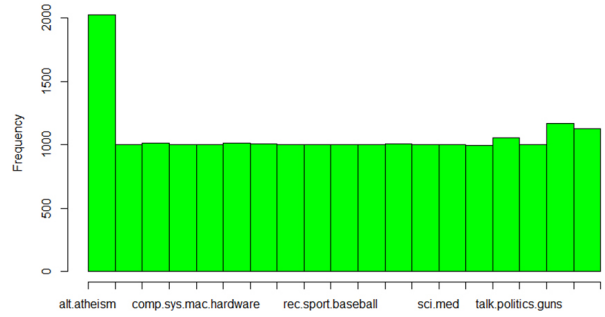

Figure 7. Category distribution for the 20 Newsgroup corpus

4. Experimental Setup

As it is previously mentioned we trained and tested more than 500 models using different combination of corpora, term weighting schemes, feature selection methods, number of attributes and classifiers. In our experiments we used 5 corpora, 3 term weighting schemes, 3 feature selection methods, 7 attribute thresholds and 5 classifiers. The total number of resulting combinations was 1,575, but we only selected a subset of them. Initial experiments were conducted with all the features and weighting schemes. Later experiments with feature selection used one weighting scheme only as we did not find any statistically significant difference among the performance of the weighting schemes. Also decision tree was dropped, because of excessively large training times.

All experiments were conducted using RapidMiner Studio 5[11] with the Text Processing and Weka plugins installed on a single workstation with 2 Intel Xeon X5650, 6 core, 2.67 GHz processors and 48 GB of memory.

4.1 Text preprocessing

The tokenized text was stemmed using Porter stemmer and stopwords were removed. Table 3 displays the corpus statistics after preprocessing. Figure 8 displays the TTR after stemming and stopword removal. Except for the movies corpus, the TTR for all other bigger corpora was observed to remain the same before and after text preprocessing. For each corpus, three document term matrices were created using TFIDF, TF and boolean term weighing schemes.

Table 3. Corpus statistics after stemming and stopword removing

Corpus	Types	Tokens	TTR
Movies	25,236	722,372	3.49%
20 Newsgroups	70,521	3,028,793	2.33%
Reuters-21578	23,985	1,372,505	1.75%
Ohsumed	41,462	6,197,219	0.67%
IMDB movie reviews	70,603	6,441,469	1.10%

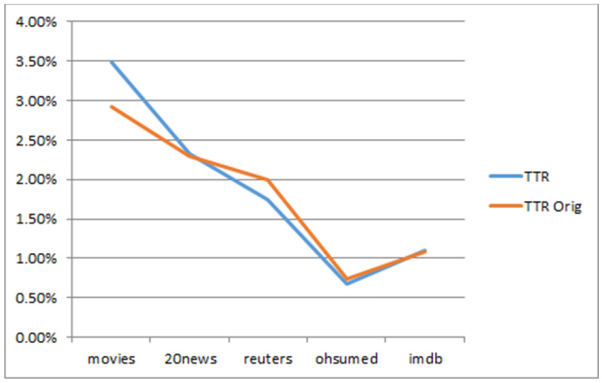

Figure 8. TTR before and after stemming and stopword removal

4.2 Feature selection methods

We used three feature selection methods, namely, Information Gain (IG), Chi Square (Chi) and Document Frequency (DF) to rank the attributes. For each method, seven weight thresholds were identified to select the top 1%, 2%, 3%, 4%, 5%, 10%, 15% and 20% attributes.

4.3 Classifiers

Five classifiers from different categories were used in our experiments. These include Support Vector Machines (discriminative) from the LibSVM package implemented in RapidMiner, Naïve Bayes (generative), C4.5 Decision Tree (non parametric) implemented in Weka, K-Nearest Neighbor (lazy, instance based) implemented in RapidMiner and Random Forest (bagging) implemented in Weka. Because of the size of the experiment, extensive parameter tuning was not possible, but limited parameter tuning was performed. SVM was used with a linear kernel, KNN was used with 50-100 neighbors with Euclidean distance computation, random forest was built with 100 trees and decision tree was built with the default settings provided by RapidMiner. The parameter values were determined empirically using pilot runs. Since SVM is inherently a binary classifier, the LibSVM package implements a one-vs-all approach to deal with a multiclass problem.

4.4 Performance measures

There are different measures available to determine the performance of a binary classification problem. The most common among them are accuracy, precision, recall and F measure. For multiclass problems in text categorization, a macro or micro averaging is performed to obtain a single measure. In the former, the performance measure for binary decisions on each individual category (ci, ¬ci) is computed and then averaged over all the categories. In the latter case, the performance measure is averaged over all the m $\times$ n binary decisions globally, where m is the number of test documents and n is the number of categories. RapidMiner computes the micro-averaged F measure and this is the performance measure reported for our experiment. We used the micro-averaged F measure as this has been reported for cross method comparison by other researchers.[2] In the rest of the paper, the micro-averaged F measure will simply be referred to as F measure.

4.5 Validation

Because of the sheer number of classifiers built, hold out method was used to validate each classifier with the training and testing ratio set to 70/30. An n-fold cross validation method (n $\geq$ 3) would have been more reliable but with the available resources, this was not an option.

5. Results and Discussion

Results were tested for statistical significance using non-parametric tests including the Kruskal-Wallis and Wilcoxon signed rank test. It was observed that the F measure was not normally distributed, which could be attributed to the presence of low skew and high skew corpora resulting in a bi-modal distribution. We will present the results of the normality test for the term weighting schemes only.

5.1 Term weighting schemes

We performed a pilot study to determine the effect of term weighting scheme on classifier performance. We only included three corpora in this study and investigated the effect of the three weighting schemes: TFIDF, TF and Boolean on classifier performance. We presented the null hypothesis that there is no difference among the means of the F measure achieved using the three term weighting schemes.

To test the hypothesis, we trained and tested 18 models for each term weighting scheme with different combinations of three corpora and six classifiers. No feature selection was applied at this stage. To choose the most appropriate statistical test, we first plotted the histograms as displayed in Figures 9-11. The histograms show that the data is not normally distributed as it included the F measures for both low-skew and high-skew corpora resulting in a bi-modal distribution. For further confirmation, Shapiro-Wilk normality test was performed and the test statistic W and p-values are displayed in Table 4. The null hypothesis that the data were normally distributed was rejected for each case as the p-values were less than the chosen significance level of .05.

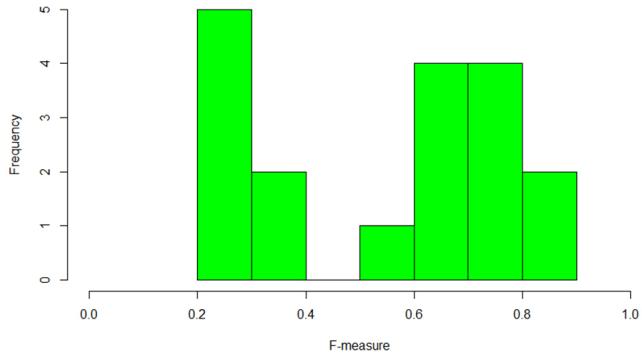

Figure 9. Distribution of F measure for TFIDF input

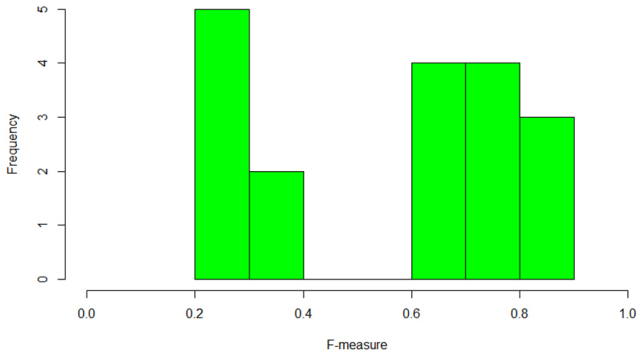

Figure 10. Distribution of F measure for TF input

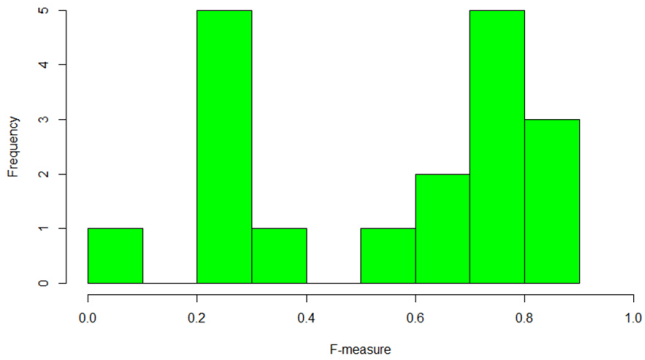

Figure 11. Distribution of F measure for Boolean input

Table 4. Results from Shapiro-Wilk test for normality

Type	W	*p*-value
TFIDF	0.8502	.008529
TF	0.81715	.002681
Boolean	0.83809	.005527

We used a Kruskal-Wallis test in this case to determine if the difference of means is statistically significant among the three weighting schemes. The p-value obtained from the test was .9533, thus we failed to reject the null hypothesis that there is no statistically significant difference among the three means and concluded that TF, TFIDF and Boolean representations resulted in similar performance.

5.2 Feature selection methods

For each feature selection method, classifier and corpus combination, we trained and tested eight models by adjusting the feature selection thresholds to select the top 1%, 2%, 3%, 4%, 5%, 10%, 15%, 20% and 100% (all) features. Figure 12 displays the F measure averaged over the feature selection methods, classifiers and corpora. It can be observed that even though the best performance in terms of F measure was achieved using all the features, the difference is not too pronounced using smaller subsets containing a fifth of the total or even smaller number of attributes. This is a reaffirmation of what was reported earlier by other researchers.[1,4,5] It is an important factor to consider, as it will reduce the training time for some classifiers by a large amount as discussed in section 5.5. To determine if this difference was statistically significant or not, we compared the F measures obtained with all the features to the maximum F measure obtained with any subset. The data was not normally distributed as affirmed by Shapiro-Wilk test, histogram and QQ plots therefore we used Kruskal-Wallis test for the null hypothesis that the difference of means is not statistically significant. Based upon the obtained p-value of .8887, we failed to reject the null hypothesis. This is an important result as it indicates

that the difference of performance between a subset of top 20% features or less and the full feature set is not statistically significant. In other words, choosing a subset of top 20% or less of the attributes can give similar performance as using the entire feature set in a text categorization task.

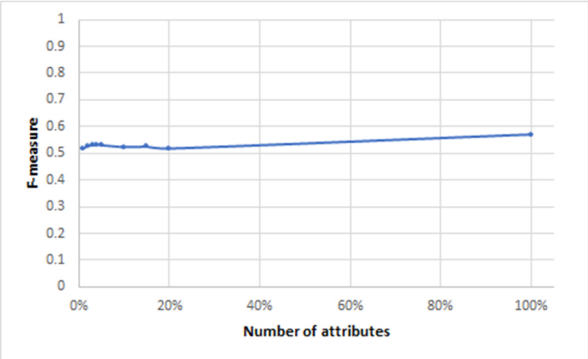

Figure 12. Average F measure against number of attributes

A comparison of feature selection methods is displayed in Figure 13. It can be seen that Chi performed better for aggressive feature selection below 7% of the original number of attributes. Once the number of attributes was increased from that threshold, IG took over. Our results are slightly different from Forman[4] in this aspect, as they reported IG to outperform Chi for smaller feature sets. Nevertheless both IG and Chi performed better than the simpler DF method as reported earlier by Yang, Pedersen and Forman.[1,4]

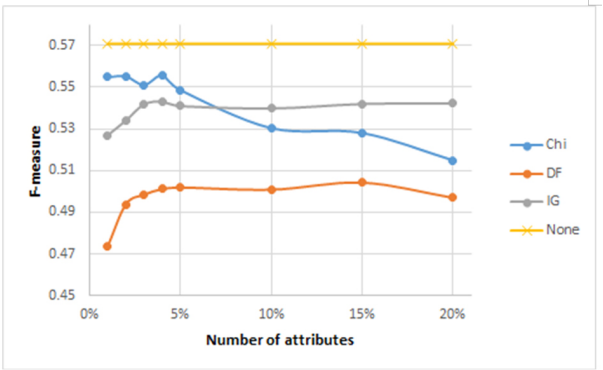

Figure 13. Average F measure against number of attributes for each feature selection method

Figure 14 displays the F measure for each corpus and feature selection method averaged over all feature subsets and classifiers. The low F measure for Ohsumed and Reuters can be attributed to the high class skew present in these corpora or to the fact that these are multi-label corpora.[2] Chi and IG produced similar results for low skew corpora, while for the high skew Chi resulted in better performance. To determine if these results were statistically significant, we performed a Kruskal-Wallis test at a significance level of 0.05 with the null hypothesis that there is no difference among the performance of these feature selection methods. With the obtained test statistic and the subsequent p-value of .13, we failed to reject the null hypothesis and concluded that even though Chi and IG performed better than DF, the difference in the performance was not statistically significant.

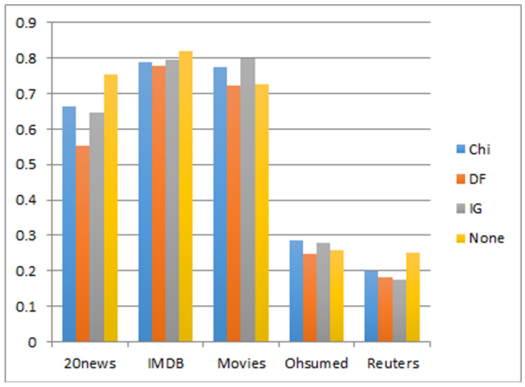

Figure 14. Average F measure for each feature selection method against each corpus

5.3 Classifiers

Our experiment reaffirmed the superiority of SVM in text categorization over the other classifiers as it has been discussed before by Yang, Liu and Forman.[2,4] Figure 15 displays the F measure for each classifier against the number of attributes averaged over all the corpora and feature selection methods. As the number of attributes was increased, SVM, NB and RF displayed similar behavior, while KNN was more peculiar. As the number of attributes was increased from 1% to 10%, a performance increase was observed for the first group of classifiers. Further increase, from 10% to 20%, did not result in any discernable change. For KNN the performance decreased when the number of attributes was increased from 1% to 20%. Further increase for 25%, 50% and 75% features was tested for the small Movies corpus only. Results indicated a further decline in performance till 25% features before an improvement was observed. For full feature set, NB displayed a slight decrease in performance while the other reported an increase or no change. The decrease in NB performance could be attributed to its class conditional feature independence assumption. As more features are added, multicollinearity tend to increase. It should be noted that there are no data points between 20% and 100% features and the curve is interpolated to show the difference of performance between subsets and full feature set.

ISSN 1927-6974 E-ISSN 1927-6982

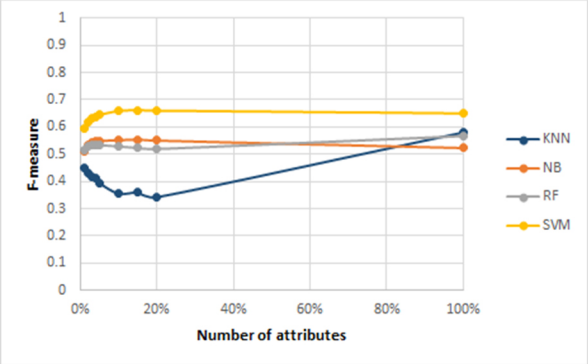

Figure 15. Average F measure against the number of attributes for each classifier

Figure 16 breaks the performance to low and high skew corpora to provide further insight. SVM proved to be the best classifier for both low and high skew corpora as affirmed by Forman,[4] followed by RF, NB and KNN. It can be observed that KNN performed better than NB and RF for high skew corpora.

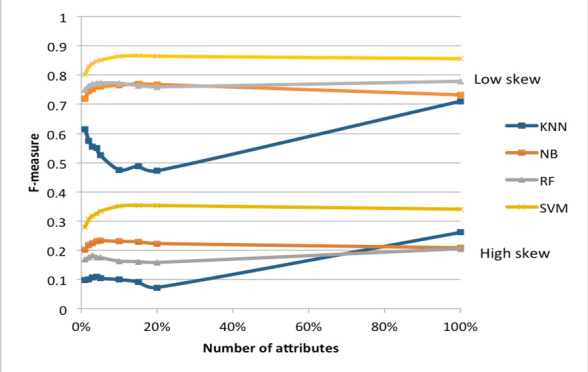

Figure 16. Average F measure against the number of attributes for each classifier for the low and high skew corpora

To compare the performance with and without feature selection for each classifier, we identified the maximum performance achieved by each classifier for any feature subset and feature selection method. This maximum is listed against the performance achieved without feature selection for each classifier in Table 5. The first column for each classifier (marked by superscript w) contains the maximum F measure achieved with feature selection while the second column (marked by superscript wo) contains the F measure without feature selection. Since we wanted to compare the performance of each classifier with and without feature selection, we conducted a series of two tailed Wilcoxon signed rank test at a significance level of 0.05, instead of using the Kruskal-Wallis test. The latter is used for an overall comparison of three or more groups. The p-values obtained from the tests were .1875, .3125, .1875 and .1875 for KNN, NB, RF and SVM comparisons respectively. Based upon the p-value for each test, we failed to reject the null hypothesis that there is no difference between the performance with and without feature selection.

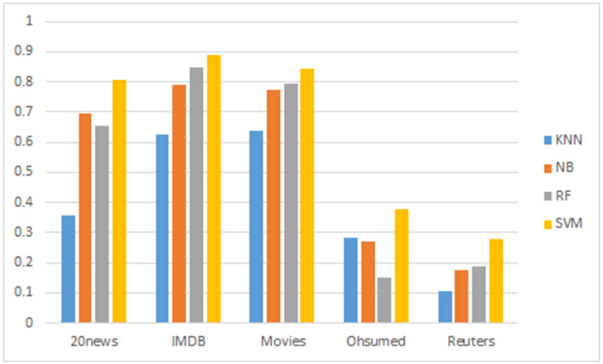

Figure 17. Average F measure for each classifier and for each corpus

A comparison of classifiers, plotted for each corpus, is provided in Figure 17. Once again, it can be seen that SVM performed the best. It is important to mention that some authors[2,3] reported much higher micro-averaged F measure for Reuters 21758 and Ohsumed corpora, but their results were provided for the top categories only, that contains a significant number of training documents. Our results are reported for all the categories in these corpora, irrespective of their size. Also in high skew corpora, micro-averaging favors common categories, therefore in the presence of under-represented categories, the value is much lower. To determine if these result of classifier comparison were statistically significant, we performed a series of one tailed Wilcoxon signed rank test for each classifier pair at 0.05 significance level. The one tailed test was preferred over two tailed test as we were interested in finding out which classifier performed better rather than finding out if there was a difference or not. The Wilcoxon tests revealed statistically significant difference between classifiers performance which can be summarized as SVM > (NB, RF) > KNN. Table 6 displays the comparison for each pair of classifiers. For each pair of classifiers, we put forth the null hypothesis that classifier 1 performed better than classifier 2. The < in Table 6 indicates that the null hypothesis was rejected in favor of classifier 2, while $\sim$ means that no statistically significant difference was observed. The test statistic V, and the p-value for each test is also provided.

Table 5. Average F measure for each classifier with and without feature selection on each corpus

Corpus	KNN^{w}	KNN^{wo}	NB^{w}	NB^{wo}	RF^{w}	RF^{wo}	SVM^{w}	SVM^{wo}
Movies	0.73	0.69	**0.90**	0.69	0.84	0.77	0.89	0.82
20 Newsgroups	0.60	0.73	0.77	0.78	0.76	0.78	0.85	**0.86**
Reuters 21578	0.19	0.26	0.21	0.22	0.22	0.21	**0.34**	0.32
Ohsumed	0.37	0.39	0.31	0.20	0.20	0.08	**0.40**	0.36
IMDB	0.69	0.83	0.82	0.73	0.86	0.83	**0.90**	0.89

Table 6. Classifier comparison result

Classifier1	Classifier2	V	*p*-value	Result
KNN	NB	566	2.2e-16	<
KNN	RF	1,095	2.087e-12	<
KNN	SVM	10	2.2e-16	<
NB	RF	4,141	0.6924	~
NB	SVM	4	2.2e-16	<
RF	SVM	99	2.2e-16	<

Figure 18 displays the F measure for each classifier and feature selection method averaged over all the corpora and feature subsets. On average, SVM performed best with Chi, but IG was not too far behind. The distribution of F measure for each classifier is displayed in Figure 19. The bi-modal nature of the F measures is evident as it has been mentioned previously. SVM's peak at the far right indicates the best performance. Even for the high skew corpora, where F measures were between 0.1 and 0.4, SVM reported the best performance. KNN was the lowest performer in our experiments, averaging at 0.6, even for the low skew corpora.

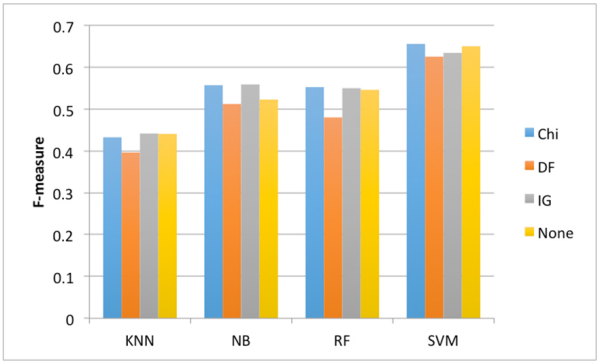

Figure 18. Average F measure for each classifier and feature selection method

5.4 Categories

While it is already established that classifiers resulted in better performance for low skew corpora versus high skew corpora, we investigated the effect of the number of categories on performance also. Figure 20 displays the F measure averaged over classifiers, feature selection methods and feature subsets against the number of categories. Figure 21 breaks this down for each classifier. The sudden drop in the F measure when the number of categories is increased form 20 to 23 is attributed to the high skew.

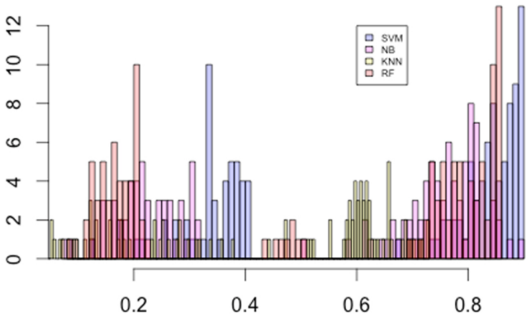

Figure 19. Distribution of the F measure for each classifier

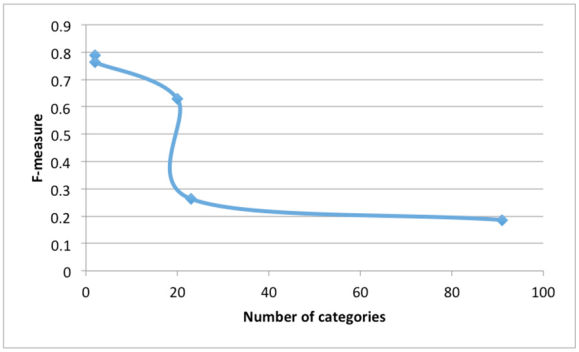

Figure 20. Average F measure against number of categories

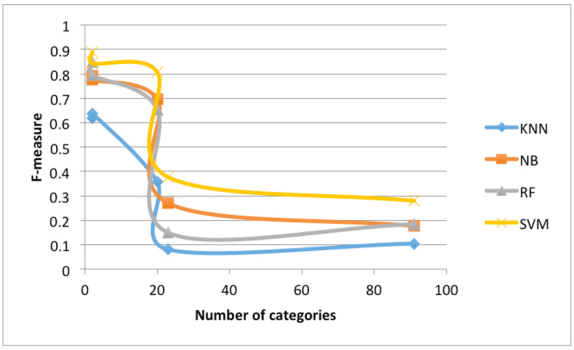

Figure 21. Average F measure against number of categories for each classifier

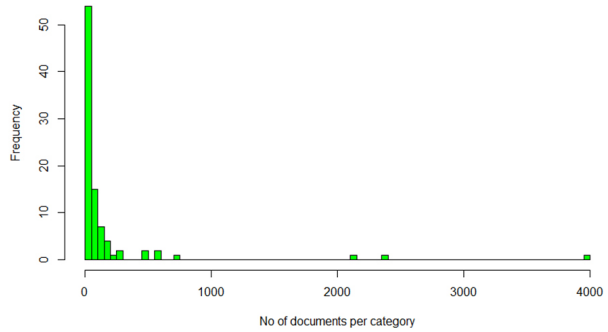

Figure 22. Distribution of the number of documents per category for the Reuters 21,578 corpus

KNN, NB and RF can handle multiple classes while SVM is inherently a binary classifier. RapidMiner uses a one-vs-all strategy for the LibSVM package to deal with multiple classes. Other strategies to deal with multiclass classification includes one-vs-one or more sophisticated techniques as reported by Constantinidis and Andreadis.[12] Multiclass classification is considered a harder problem than binary classification for two main reasons: 1) to effectively learn each class, a certain number of examples per class is required resulting in a larger number of overall training examples and 2) class imbalance, if present, will become more pronounced as there will be more underrepresented classes. The presence of multi-label classification makes the problem even harder as a single document may belong to different classes.

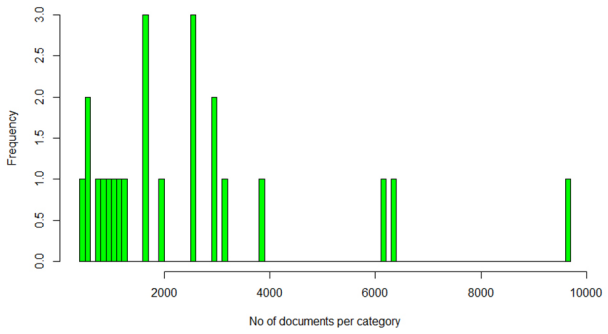

Figure 23. Distribution of the number of documents per category for the Ohsumed corpus

Table 7. Descriptive statistics of the number of documents per categories in each corpus

Corpus	Mean	St Dev	Min	Max
Movies	1,000	0	1,000	1,000
20 Newsgroups	1,021	45.56	997	1,171
Reuters 21578	169.6	530.59	2	3,964
Ohsumed	2,478	2,225.12	427	9,611
IMDB	25,000	0	25,000	25,000

It was reported by Yang and Liu[2] that Reuters 21578 contained 1.2 while Ohsumed contained 12 to 13 categories per document making categorization for the latter a more difficult task. The distribution of the number of documents per category for the Reuters 21578, Ohsumed and 20 Newsgroups corpora is displayed in Figure 22, Figure 23 and Figure 24 respectively. The distributions have a right skew indicating the presence of a few outlying categories with a large number of documents belonging to them. The descriptive statistics are provided in Table 7. The mean number of documents per category in Reuters 21578 is about 170 while there are few categories with more than 500 documents in them. Similar pattern can be found in Ohsumed and 20 Newsgroups corpora. Figure 25 can be interpreted in light of this that the sudden drop in F measure while going from 20 to 23 categories may not have a lot to do with just the number of categories but a consequence of multiclass classification with imbalance classes in the latter case.

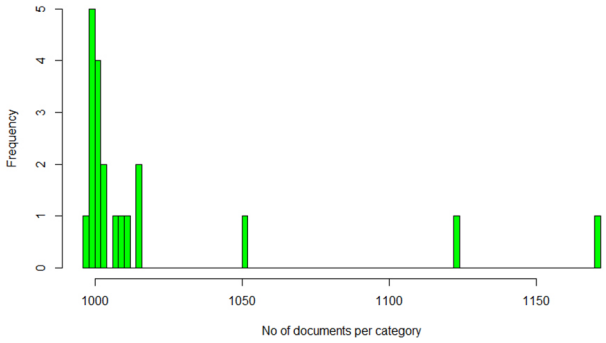

Figure 24. Distribution of the number of documents per category for the 20 Newsgroups corpus

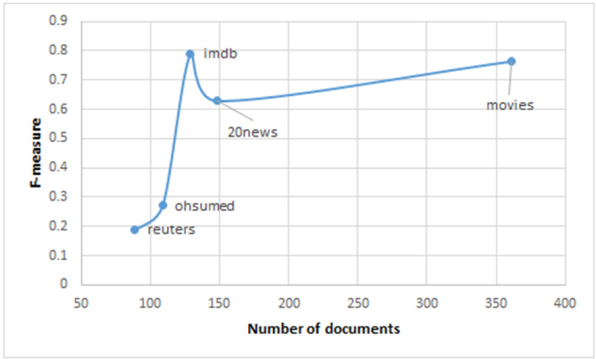

Figure 25. Average F measure against number of documents

5.5 Document length

To investigate the effect of document length on F measure, we plotted the latter against the average document length for each corpus. While an increasing trend can be observed, it would be premature to attribute it to the document length only, especially when the document length is tied to the corpus. The correlation between the F measure and average

document length was 0.5437, but to establish a cause and effect relationship other, corpus specific factors, such as category distribution, number of categories, multi-labeling, have to be taken into account.

5.6 Classifier training time

A direct consequence of data complexity in text categorization problems is large training time. Complexity arises as a direct consequence of high dimensionality, multiple categories and large number of available documents to process. Figure 26 displays the training time as a function of the number of attributes, averaged over corpora and feature selection methods for each classifier. While NB reported the smallest training times as it involves simple probability computations, the training times for SVM were comparably low also. RF being a tree-based classifier suffers from high dimensionality when it comes to training times, even though it works with a subset of attributes at a time. As it was previously mentioned, we grew the forest of 100 trees. KNN reported the longest training times. It is important to mention that these are actually the model application time as KNN is a lazy classifier and actual classification is delayed till the presentation of a test instance. Since the test instance has to be compared against all the instances present in the data, the classification time increases with the number of data instances. Also, it should be noted that there were no data points between 20% to 100% attributes and the figure displays interpolated values. Training times for each corpus are displayed in Figure 27. Ohsumed and IMDB being the largest corpora in terms of the number documents, reported highest training times, especially for KNN.

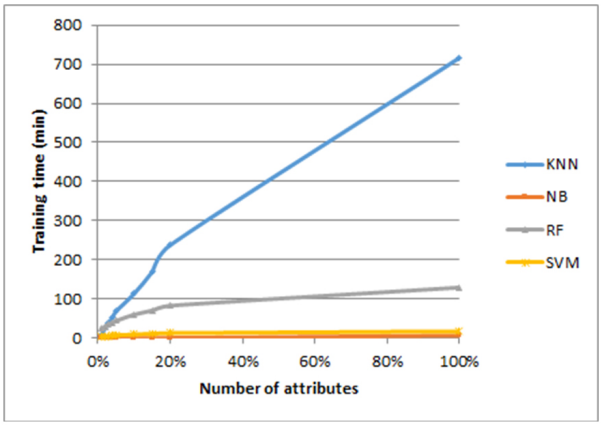

Figure 26. Average training time as a function of the number of attributes

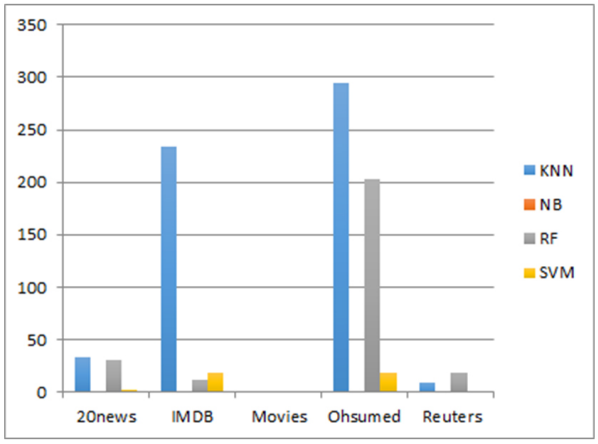

Figure 27. Average training time for each corpus

To get the overall picture, Figure 28 displays the distribution of the training time for each classifier. The outliers resulting in the right skew are attributed to the KNN classification times. Figure 29 displays the distribution of the training time with outliers removed by limiting the training times to 100 minutes maximum only. It can be noted that the maximum training times posted by SVM and NB were under 40 minutes.

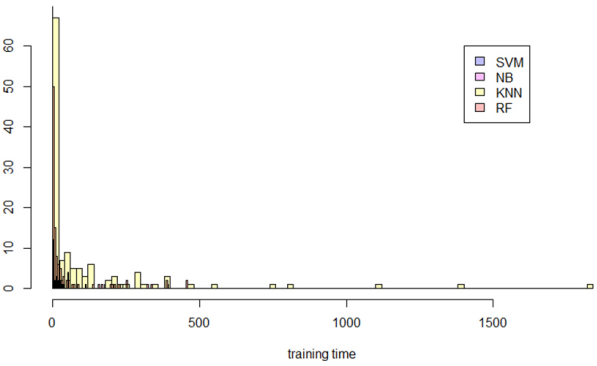

Figure 28. Distribution of the training time for each classifier

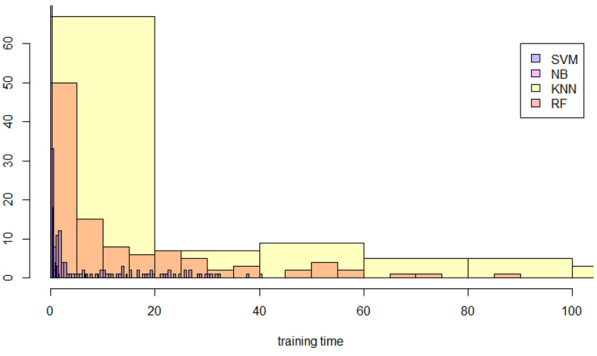

Figure 29. Distribution of the training time (limited to 100 minutes) for each classifier

6. CONCLUSION

We presented an experimental evaluation of feature selection methods and classifiers for text categorization. Our results proved the supremacy of SVM with linear kernel for text categorization tasks on large corpora, both in terms of higher F measure and lower training time. RF and NB gave almost similar performance, but the former had much longer training times. If training time is not an issue, RF can be tuned to perform better using a higher number of trees (100 used in our experiment) and tinkering with the number of features to sample for a single tree in the forest. Attribute discretization also helps in reducing training time. KNN performed worst in our experiments, both in terms of F measure and training time. The latter were prohibitively high for larger corpora.

REFERENCES

[1] Yang Y, Pedersen J. A comparative study of feature selection in text categorization. Fourteenth International Conference on Machine Learning. ICML. San Francisco; 1997.

[2] Yang Y, Liu X. A re-examination of text categorization methods. The 22nd annual international ACM SIGIR conference on Research and development in information retrieval: New York; 1999.

[3] Yang Y. An Evaluation of Statistical Approaches to Text Categorization. Information Retrieval. 1999; 1(1-2): 69-90.

[4] Forman G. An Extensive Empirical Study of Feature Selection Metrics for Text Classification. The Journal of Machine Learning Research. 2003; 3: 1289-1305.

[5] Lewis D. Feature Selection and Feature Extract ion for Text Categorization in Workshop on Speech and Natural Language HLT 91, Stroudsburg; 1992.

[6] Lewis D. Evaluating Text Categorization in Workshop on Speech and Natural Language, HLT 91, Stroudsburg; 1991.

[7] Aggarwal C, Zhai C. A Survey of Text Classification Algorithms in Mining Text Data, Boston, Springer US; 2012. p. 163-222.

[8] Sebastiani F. Machine Learning in Automated Text Categorization. ACM Computing Surveys. 2002; 34(1): 1-47. http://dx.doi.org/10.1145/505282.505283

[9] Khan A, Baharudin B, Hong L, *et al*. A Review of Machine Learning Algorithms for Text-Documents Classification. Journal of Advances In Information Technology. 2010; 1(1): 4-20.

[10] Moschitti A. Text Categorization Corpora [Internet]. 2004. Available from: http://disi.unitn.it/moschitti/corpora.htm. [Accessed 23 12 2016].

[11] Hofmann M, Klinkenberg R. RapidMiner: Data Mining Use Cases and Business Analytics Applications, Chapman & Hall/CRC; 2013.

[12] Constantinidis I, Andreadis I. Partitioning trees: A global multiclass classification technique for SVMs. Artificial Intelligence Research. 2014; 3(2): 41-56. http://dx.doi.org/10.5430/air.v3n2p41