A comparative study of selection and machine learning techniques for sentiment analysis

Conference Paper · October 2012

DOI: 10.1145/2401603.2401605

2 authors:

Anuj Sharma
Indian Institute of Management Indore
28 PUBLICATIONS **856** CITATIONS

SEE PROFILE

Shubhamoy Dey
Indian Institute of Management Indore
57 PUBLICATIONS **721** CITATIONS

SEE PROFILE

A Comparative Study of Feature Selection and Machine Learning Techniques for Sentiment Analysis

Anuj Sharma
Indian Institute of Management
Prabandh Shikhar, Rau
Indore – 453331, India
f09anujs@iimidr.ac.in

Shubhamoy Dey
Indian Institute of Management
Prabandh Shikhar, Rau
Indore – 453331, India
shubhamoy@iimidr.ac.in

ABSTRACT

Sentiment analysis is performed to extract opinion and subjectivity knowledge from user generated text content. This is contextually different from traditional topic based text classification since it involves classifying opinionated text according to the sentiment conveyed by it. Feature selection is a critical task in sentiment analysis and effectively selected representative features from subjective text can improve sentiment based classification. This paper explores the applicability of five commonly used feature selection methods in data mining research (DF, IG, GR, CHI and Relief-F) and seven machine learning based classification techniques (Naïve Bayes, Support Vector Machine, Maximum Entropy, Decision Tree, K-Nearest Neighbor, Winnow, Adaboost) for sentiment analysis on online movie reviews dataset. The paper demonstrates that feature selection does improve the performance of sentiment based classification, but it depends on the method adopted and the number of feature selected. The experimental results presented in this paper show that Gain Ratio gives the best performance for sentimental feature selection, and SVM performs better than other techniques for sentiment based classification.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Information filtering.* I.2.7 [**Natural Language Processing**] – *Text analysis.*

General Terms

Performance, Measurement, Experimentation.

Keywords

Feature Selection, Sentiment Analysis, Text Classification.

1. INTRODUCTION

With the rapidly increasing amount of user generated text available on the internet, organizing this vast amount of unstructured text data into structured information has become increasingly important. Data mining or more specifically, text mining techniques are used to extract knowledge from this type of user generated text content. The extracted knowledge can be utilized for different exploratory or predictive analysis purposes. Sentiment analysis (often referred as opinion mining) is a recent area of research where we apply advanced text mining (TM), machine learning (ML), information retrieval (IR) and natural language processing (NLP) approaches to process vast amounts of user generated text content. Sentiment analysis is performed to extract the opinion and subjectivity knowledge from online text, formalize this knowledge discovered and analyze it for specific use [15].

Sentiment analysis may be as simple as basic sentiment based categorization of text documents, to more complex procedures to extract opinion at different granularity levels [19]. Sentiment analysis is the task of automatically judging the sentiment orientation (positive or negative) of subjective text. The application of this can be classifying online product reviews as positive or negative to judge the product as recommended or not recommended [34]. Sentiment analysis can also be employed to process standpoint, view, and mood of the public for political concerns as a part of a public opinion analysis system. Other applications may be in online question–answer systems and opinionated text summarization [26].

Sentiment based classification of text documents is a more challenging tasks than topic based classification. Discrimination based on opinions, feelings, and attitudes is a more complex task than classification based on topics. The opinionated or subjective text on the Web is often non-structured or semi-structured from which feature selection is a crucial problem [8, 26, 32]. Moreover, the sentiment features are not expressed objectively and explicitly, and usually are hidden in a large pool of subjective text. Therefore, the text sentiment classification requires deeper analysis and understanding of textual features. In this research work, the issue of feature selection for sentiment based classification of text documents (specifically online reviews) is addressed. This paper presents an empirical comparison of feature selection methods for sentiment analysis and attempts to find the synergy between feature selection methods and machine learning based classifiers.

Five traditional feature selection methods i.e., information gain (IG), gain ratio (GR), CHI statistics (CHI), Relief-F and document frequency (DF) are experimented with in this study. The classification performance of the feature selection methods is investigated using seven machine learning classifiers, i.e., K-nearest neighbor (KNN) [29], , Naïve Bayes (NB) [16], Winnow [33], Maximum Entropy (ME) [2], Decision Tree (C4.5) [22], Adaboost [9] and support vector machine (SVM) [11]. We have conducted experiments on movie review dataset comprising reviews of movies from the Internet Movie Database (IMDb) adopted from Pang and Lee [18, 20].

RACS'12, October 23-26, 2012, San Antonio, TX, USA.
Copyright 2012 ACM 978-1-4503-1492-3/12/10…$15.00.

The rest of this paper is organized as follows: Section 2 presents related work on sentiment analysis. Feature selection and machine learning techniques are described in Section 3. Experimental results are given in Section 4. Finally Section 5 concludes this paper.

2. RELATED WORK

There are some comprehensive reviews available in research literature related to sentiment analysis [19, 25] that describe different techniques used for sentiment analysis in text documents at different level (i.e. document, sentence and feature level).

The popular approaches to sentiment analysis are based on machine learning techniques, semantic analysis techniques, statistical techniques and lexicon or dictionary based techniques. This paper strictly limits its scope to work related to applying machine learning classification techniques and feature selection methods. Table 1 describes some of the studies related with sentiment based classification of text documents using machine learning classifiers. Machine learning-based approaches aim at finding patterns from pre-coded text documents or snippets during learning and uses multi fold cross validation for assessment of the accuracy of the built models.

Different machine learning classifiers like Naïve Bayes (NB) [1, 3, 4, 7, 12, 18, 21, 28, 31], Support Vector Machine (SVM) [7, 10, 18, 20, 21], Maximum Entropy (Max Ent) [3, 7, 21, 23], Winnow classifier [5], Decision Trees (C4.5) [1, 4] have been used extensively for classification model building. The other classifiers like K-nearest neighbour (KNN) [24] and Adaboost [27] have also been studied for sentiment analysis in different domains.

Extracting the best features in appropriate numbers is an open issue in machine learning-based sentiment classification. Different studies have tried to resolve the issues relating to extraction of complex features and compared feature selection methods [21]. Most of the existing research focus on simple features, including single words [24], character N-grams [32], word N-grams [10, 18] like bigrams and trigrams [5, 7], or the combination of aforementioned features. Some other studies have adopted different feature selection methods like log likelihood tests [4], fisher's discriminant ratio [26], information gain and CHI statistics [24]. Though some of the feature selection methods mentioned above have been studied individually, there are few studies that have compared and analyzed the performance of different types of features selection methods, when used with different machine learning techniques [30].

3. METHODOLOGY

This section presents the method of sentiment analysis used in this study. First, the review documents were collected and pre-processed with basic natural language processing techniques like word tokenization, stop word removal and stemming. The residual tokens were arranged as per their frequencies or occurrences in the whole documents set. Then different feature selection methods were utilized to pick out top n-ranked discriminating attributes for training the classifiers. The number of selected features (n) was varied from very small to very large (100 – 10000). Seven machine learning based classifiers were applied to evaluate the effectiveness of different feature selection methods on the basis of performance of the sentiment analysis task.

Table 1. Research Work Related to Machine Learning Classifiers for Sentiment Analysis

Author	Model	Data Source and Dataset	Accuracy (%)
Pang et al. (2002) [21]	NB, ME, SVM	Movie reviews (IMDb)- 700 (+) and 700 (-) reviews	77–82.9
Dave et al. (2003) [7]	NB, ME, SVM	Product reviews (Amazon)	88.9
Pang & Lee (2004) [18]	NB, SVM	Movie reviews (IMDb)- 1000 (+) and 1000 (-) reviews	86.4-87.2
Gamon (2004) [10]	SVM	Customer reviews (feedback)	69.5-77.5
Pang & Lee (2005) [20]	SVM, SVR, Regression, Metric Labeling	Movie reviews (IMDb)- 5006 reviews	54.6-66.3
Cui et al., (2006) [5]	Winnow, Discriminative ML classifier	Online electronic product reviews- 320k reviews	F1 Score-0.90
Kennedy & Inkpen (2006) [13]	SVM	Movie reviews (IMDb)- 1000 (+) and 1000 (-) reviews	80– 85.9
Chen et al. (2006) [4]	Decision Trees C4.5, SVM, NB	Books Reviews (Amazon)- 3,168 reviews	84.59
Boiy et al. (2007) [3]	SVM, Multinomial NB, ME	Movie reviews (IMDb)- 1000 (+) and 1000 (-) reviews, Car reviews- 550 (+) and 222 (-) reviews	90.25
Annett & Kondrak (2008) [1]	SVM, NB, Decision Tree	Movie reviews (IMDb)- 1000 (+) and 1000 (-) reviews	Greater than 75%
Shimada & Endo (2008) [23]	SVR, SVM OVA, ME	Product Reviews (video games)	NA
Dasgupta & Ng (2009) [6]	SVM and Clustering based	Movie reviews (IMDb) and product reviews (Amazon)- 1000 (+) and 1000 (-) reviews	69.5-93.7
Ye et al. (2009) [31]	NB, SVM and Character based N-gram model	Travel blogs from travel.yahoo.com- 591 (-) and 600 (+) reviews	80.71-85.14
Paltoglou & Thelwall (2010) [17]	SVM	Movie Reviews (IMDb)- 1000 (+) and 1000 (-) reviews, Multi-Domain Sentiment Dataset (MDSD)- 8000 reviews	MR-96.90, MDSD-96.40
Xia et al. (2011) [28]	NB, ME, SVM, meta-classifier combination	Movie Reviews (IMDB), product reviews (Amazon)- 1000 (+) and 1000 (-) reviews	88.65
Kang et al. (2011) [12]	Improved NB	Restaurant Reviews- 5700 (+) and 757 (-) reviews	83.6

3.1 Feature Selection Methods

Feature selection methods reduce the original feature set by removing irrelevant features for text sentiment classification to improve classification accuracy and decrease the running time of learning algorithms. We have investigated performance of five commonly used feature selection methods in data mining research, i.e., DF, IG, CHI, GR and Relief-F. All these feature selection methods are used to compute a score for each individual feature and then a predefined number of features are selected as per ranking obtained from that score.

3.1.1 Document Frequency (DF)

Document Frequency measures the number of documents in which the feature appears in a dataset. This method removes those features whose document frequency is less than or greater than some predefined threshold frequency range. Selecting frequent features may increase the probability that the features will be present in future evaluation test cases. The basic assumption is that both rare and common features are either non-informative for sentiment category prediction, or not impactful in improving classification accuracy [30]. Research literature shows that this method is simplest, scalable and effective for text classification [24].

3.1.2 Information Gain (IG)

Information has been used frequently as a feature (term) goodness criterion in machine learning based classification [24, 26, 30]. It measures the information required (in bits) for class prediction of a document, based on the presence or absence of a feature term in that document. Information Gain is derived by accessing the impact of feature's inclusion on decreasing overall entropy. The expected information needed to classify an instance (tuple) for partition D or identify the class label of an instance document in D is known as entropy and is given by:

$$Info(D) = -\sum_{i=1}^{m} (P_i)\log_2(P_i) \tag{1}$$

Where m symbolizes the number of classes (e.g., 2 for binary classification). P_i represents the probability that an arbitrary instance document in D labelled as class C_i and is calculated as $|C_{i,\ D}|\ /\ |D|$ (proportion of tuples of each class). The log function to the base 2 confirms encoding of information in bits. If we have to classify the instance in D on some attribute $A\ \{a_1...,\ a_v\}$, D will split into v partitions set $\{D_1,\ D_2,...D_v\}$. The information, we require to turn up an exact classification is measured by:

$$Info_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j) \tag{2}$$

Where $|D_j|/|D|$ is the weight of the j^{th} partition and $Info(D_j)$ is the entropy of partition D_j. Finally Information gain by partitioning on A is:

$$Information\ Gain(A) = Info(D) - Info_A(D) \tag{3}$$

We select the attributes ranked as per the highest information gain that reduce the information required to classify the documents in the resultant classes.

3.1.3 Gain Ratio (GR)

Gain Ratio enhances Information Gain as it normalizes contribution of all features to final classification decision for a document. An iterative process selects smaller sets of features decrementally by utilizing GR score. The iterations terminate when the predefined number of features remain. Gain ratio is used as a disparity measure, and high GR score indicates that the selected features will be useful for classification. Gain Ratio was introduced in decision tree (C4.5) algorithms. The normalization score is known as split information value [22]. The split information value is expressed by the prospective information obtained by partitioning the training document set D into v separations, corresponding to v outcomes on feature A:

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2 \frac{|D_j|}{|D|} \tag{4}$$

Where high *SplitInfo* means partitions are uniform and low *SplitInfo* means few partitions hold most of the tuples (peaks). Finally the gain ratio is defined as:

$$Gain\ Ratio(A) = Information\ Gain(A) / \ SplitInfo(A) \tag{5}$$

3.1.4 CHI statistic (CHI)

The Chi Squared statistic (CHI) represents the association between the feature and the corresponding class. The divergence from the expected distribution is measured by this statistical test based upon assumption that the feature occurrence is independent of the final class value [24, 30]. It is defined as,

$$CHI(t, c_i) = \frac{N \times (AD - BE)^2}{(A+E) \times (B+D) \times (A+B) \times (E+D)} \tag{6}$$

$$CHI_{\max}(t) = \max_i(CHI(t, c_i)) \tag{7}$$

Where A is the frequency when t and C_i co-occur; B represents the number of events when t occurs without C_i. E represents the number of occasions when C_i occurs without t; D is the frequency when neither C_i nor t occurs; N is the total instance in document set. The CHI statistic will be zero if t and C_i are independent.

3.1.5 Relief-F Algorithm

The Relief-F algorithm selects feature-instances randomly, computes their nearest neighbors, and adjusts a final feature weighting vector. The adjustment gives more importance to features that better discriminate the instance document from neighbors of other classes [14]. Specifically, it tries to find a best estimate of W_f from the following probabilities to allocate as the weight for each term feature f:

$$\begin{aligned} W_f = {} & P(different\ value\ of\ f\ \mid\ nearest\ \text{instances}\ from\ different\ class) - \\ & P(different\ value\ of\ f\ \mid\ nearest\ \text{instances}\ from\ same\ class) \end{aligned} \tag{8}$$

3.2 Machine Learning Techniques for Sentiment Classification

3.2.1 Naïve Bayes (NB)

Naïve Bayes assumes a stochastic model of document generation and uses Bayes' rule. To classify as the most probable class c^* for a new document d, it computes:

$$c^* = argmax_c P(c\ |d) \tag{9}$$

The Naïve Bayes (NB) classifier uses Bayes' rule:

$$P(c \mid d) = (P(c)\ P(d \mid c))\ /\ P(d) \tag{10}$$

$P(d)$ plays no role in selecting c^*. To estimate the term $P(d|c)$, Naïve Bayes decomposes it by assuming the conditional independence of features f_i's given d's class:

$$P_{NB}(c \mid d) = \frac{P(c)(\prod_{i=1}^{m} P(f_i \mid c)^{n_i(d)}}{P(d)} \tag{11}$$

The training procedure estimates relative-frequency of $P(c)$ and $P(f_i|c)$, using add-one smoothing. The Naïve Bayes classifier's conditional independence assumption clearly does not hold in real-world situations, but it still tends to perform surprisingly well for sentiment classification [1, 3, 4, 7, 12, 18, 21, 28, 31].

3.2.2 Support Vector Machine (SVM)

Support vector machines (SVMs) are highly effective in traditional text categorization, and can outperform Naive Bayes [11]. SVM seeks a hyper-plane represented by a vector that splits the positive and negative training vectors of documents with maximum margin. The problem of finding this hyperplane can be translated into a constrained optimization problem.

The structural risk minimization principle is utilized from the computational learning theory. SVM seeks a decision surface to separate the training data points into two classes and makes decisions based on the support vectors that are selected as the only effective elements in the training set. The optimization of SVM (dual form) is to minimize:

$$\vec{\alpha}^{*} = \arg\min\left\{ -\sum_{i=1}^{n}\alpha_i + \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \left\langle \vec{x_i}, \vec{x_j} \right\rangle \right\} \tag{12}$$

$$\text{Subject to:} \quad \sum_{i=1}^{n}\alpha_i y_i = 0;\ 0 \le \alpha_{\mathrm{i}} \le \mathrm{C} \tag{13}$$

3.2.3 Maximum Entropy

Maximum entropy classification (MaxEnt, or ME) can outperform Naive Bayes in standard text classification tasks [21]. The conditional probability *P(c|d)* is estimated by following exponential form-

$$P_{ME}(c \mid d) = \frac{1}{Z(d)} \exp\left(\sum_i \lambda_{i,c} F_{i,c}(d,c) \right) \tag{14}$$

where $Z(d)$ is a normalization function. $F_{i,c}$ is a feature or class function for feature f_i and class c, which is defined as follows:

$$F_{i,c}(d,c) = \begin{Bmatrix} 1,\ \mathrm{n_i(d) > 0 \text{ and } c' = c} \\ 0 \qquad\qquad \text{otherwise} \end{Bmatrix} \tag{15}$$

Where, $\lambda_{i,c}$ are feature-weight parameters and inspection of the definition of P_{ME} shows that a large $\lambda_{i,c}$ means that feature f_i is considered a strong indicator for class c. The parameter values are set so that the MaxEnt classifiers maximize the entropy of the induced distribution while maintaining the constraints enforced by the training data. The constraint is that the projected values of the feature/class functions with respect to the model are equal to their projected values with respect to the training data [3].

The MaxEnt classifier assumes the dependence of features so it can perform better than Naïve Bayes, which assumes that features are independent. The features are seldom independent in natural language processing applications and each feature in the MaxEnt classifier is an indicator function of some property of the document [7].

3.2.4 Decision Tree

Decision trees are made by iteratively partitioning the document feature vector space into parts. At each step, the split that improves the error of the overall tree on the training data is used, and based upon this greedy strategy, the desired sized tree is made. For new data, a label is predicted by perusing the branches of the tree from the root node as per the values of the term features of the new data [22]. Decision trees are easily interpretable, as the tree structure can be represented graphically, and we can follow branches down the tree according to the input variables requiring less time to train. This paper uses the J48 classifier which implements C4.5 algorithm for learning decision trees [1, 4].

3.2.5 K-Nearest Neighbor Classifier

The K-nearest neighbor (KNN) classifier is an instance based classifier that relies on the class labels of training documents which are similar to the test document. Thus, it does not build an explicit declarative model for the class c_i [24]. An instance is classified by a similarity based vote of its neighbors, with the instance being classified to the class most common amongst its *k* nearest neighbors (*k* is a positive number). If *k* =1, then the instance is simply assigned as per the class of its nearest neighbor. Given a test document *d*, KNN finds the *k* nearest neighbors among training documents. The similarity score of each nearest neighbor document to the test document is used as the weight of the classes of the neighbor document. The weighted sum in KNN classification can be written as follows:

$$Score(d, c_i) = \sum_{d_j \in KNN(d)} sim(d, d_j)\ \delta(d_j, c_i) \tag{16}$$

The term *KNN(d)* represents the document set of *k* nearest neighbors of document *d*. If d_j belongs to c_i, $\delta(d_j,c_i)$ equals 1, or otherwise 0. A test document *d*, should belong to the class that has the highest resulting weighted sum.

3.2.6 Winnow classifier

Winnow is an online mistake-driven classifier for learning a linear model from labeled examples. Winnow uses a multiplicative scheme for updating its weights in a sequence of trials [5]. For an instance document *d,* class prediction is done during each training trial. After receiving feedback for *d*, if a mistake is discovered, weight vector are updated using the document *d*. During training with a collection of training instances, this process is repeated several times by iterating on the data [33]. The balanced winnow algorithm keeps two weights for each feature, w_{kt}^{+} and w_{kt}^{-}. For a given instance (d_{k1}, d_{k2},... ,d_{kW}), the algorithm deems the document relevant if and only if,

$$\sum_{t=1}^{w} \ w_{kt}^{+} - w_{kt}^{-} \ \ d_{kt} \ge \tau \tag{17}$$

Where τ denotes given threshold and *k* indicates the class label. The weights of the active features are updated only when a mistake is encountered. In the promotion step, following an error on a positive instance, the positive part of the weight is promoted while the negative part of the weight is demoted. The coefficient of d_{kt} in the equation above increases after a promotion.

$$w_{kt}^{+} = w_{kt}^{+} \times \alpha \text{ and } \alpha > 1 \tag{18}$$

$$w_{kt}^{-} = w_{kt}^{-} \times \beta \text{ and } 0 < \beta < 1 \tag{19}$$

In the demotion step, by contrast, the positive part of the weight is demoted, while the negative part of the weight is promoted [33].

3.2.7 Adaboost classifier

Adaboost or Adaptive Boosting is a meta-learning method that tries to build a good learner utilizing a group of weak classifiers [9]. AdaBoost is known as adaptive since successive classifiers built are tweaked to support those instance documents misclassified by previous classifiers. The appearance of each individual feature in a document probably approximately indicates the class of the document as a weak classifier.

The AdaBoost algorithm works by calling a weak classifier several times. The weak classifier may be an indicator function of a certain word. Each time a weak classifier is called, it is provided with a different distribution over the training examples. The idea of boosting is that it assigns higher probability to the part that it doesn't classify correctly, in the hope that the new weak classifier can reduce the classification error by focusing on it. The algorithm selects the weak classifiers to use, and learns the weight for each weak classifier. In the end, hypotheses, from all iterations are combined into one final hypothesis.

4. EXPERIMENT RESULTS

4.1 Datasets and Performance Evaluations

This study uses a data set of classified movie reviews prepared by Pang and Lee [18, 20]. The data set contains 1,000 positive and 1,000 negative reviews collected from Internet Movie Database (IMDb) and known as polarity dataset v2.0 or Cornell Movie Review Dataset. This dataset is used as a benchmark dataset for sentiment analysis as movie reviews have been found to be one of the most difficult domains to be categorized as per sentiment.

To evaluate the performance of sentiment classification, this work has adopted classification accuracy as an index. Accuracy in this study is used as a statistical measure of how well a sentiment based binary classification test correctly classifies a test document. The accuracy is the proportion of true results (both true positives and true negatives) to the total population.

4.2 Experimental Design

In this work we have used the Vector space model (feature vector model) for representing text documents of online reviews. Documents are represented as vectors and each dimension corresponds to a separate feature. Java based implementations on Microsoft Windows platform were used to implement all the classifiers and parameters were selected as recommended by prior research studies.

For experiments involving SVM, this work employed radial basis function kernel as it gave better performance when tested empirically. For KNN, the number of neighbors (k) was set to 13 as it gave better performance [24]. The parameters for Winnow were set as α=2 and β=0.5 and the initial weight value was fixed as 2. All experiments were validated using 10-fold cross validation in which, the whole dataset is broken into 10 equal sized sets and classifier is trained on 9 datasets and tested on remaining dataset. This process is repeated 10 times and we take a mean accuracy of all folds.

4.3 Comparison and Analysis

Tables 2 reports the best performance of five feature selection methods combined with seven machine learning methods. Figures 1–5 display the performance curves of seven machine learning methods using different feature selection methods.

Firstly, with respect to feature selection methods, Gain Ratio (GR) performed the best across almost all machine learning methods. As reported in Table 2, GR gave best average accuracy as 0.9090 with top 5000 features when used with Naïve Bayes classifier. When experimented with all other classifiers, GR gave the best classification accuracy among all feature selection methods. This indicated that GR is a good choice among feature selection methods for sentiment analysis. Research literature also supports this finding as due to using a normalized measure of a feature's contribution to a classification decision, GR outperforms Information Gain feature selection [22].

Table 2. Best Accuracy (in %) for Different Classifiers

	NB	SVM	MaxEnt	DT	KNN	Winnow	Adaboost
DF	82.85	85.15	72.65	68.8	63.15	66.05	66.35
IG	88.85	89.20	87.35	74.45	71.15	71.15	65.20
GR	90.90	90.15	88.85	75.35	75.15	73.50	65.70
CHI	88.40	89.45	87.10	73.10	70.30	69.50	65.20
Relief-F	82.50	83.40	77.95	66.30	65.25	64.85	66.90

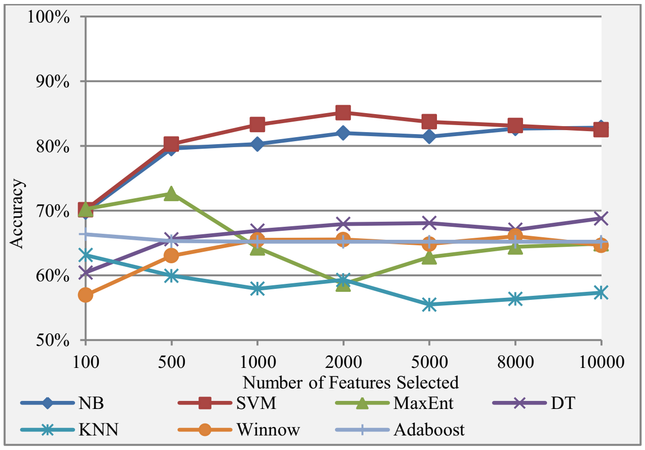

Figure 1. Performance of Machine Learning Techniques with Document Frequency Feature Selection.

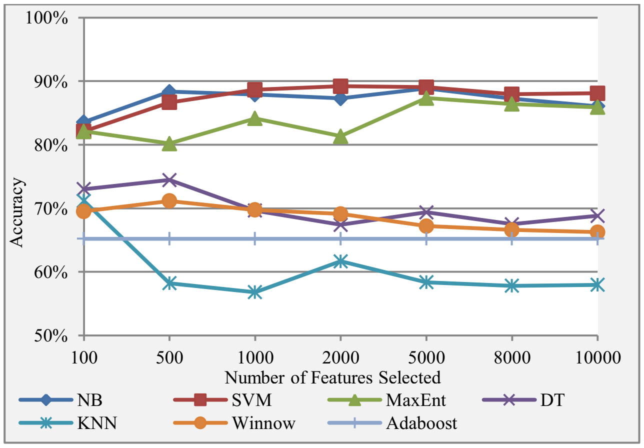

Figure 2. Performance of Machine Learning Techniques with Information Gain Feature Selection.

The basic drawback of GR found was its sensitivity to number of features selected as depicted from Figure 3. When used with very few features, GR gave poor results as compared to IG, CHI and Relief-F. Chi Squared Feature Selection method gave stable results with NB and SVM similar to Information Gain. Relief-F and DF gave poor results as compared to GR, IG and CHI. As we can see, IG is a better choice when we need stable results and do not want to experimentally determine the different number of features to be selected. This result is also consistent with a previous study that compared IG with CHI, DF and Mutual Information based feature selection methods [24].

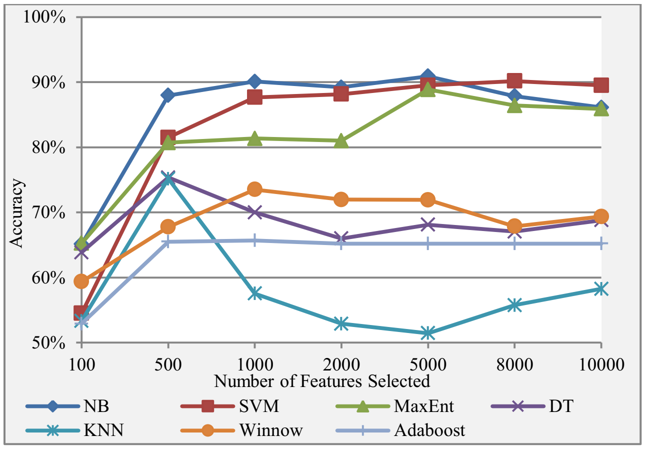

Figure 3. Performance of Machine Learning Techniques with Gain Ratio Feature Selection.

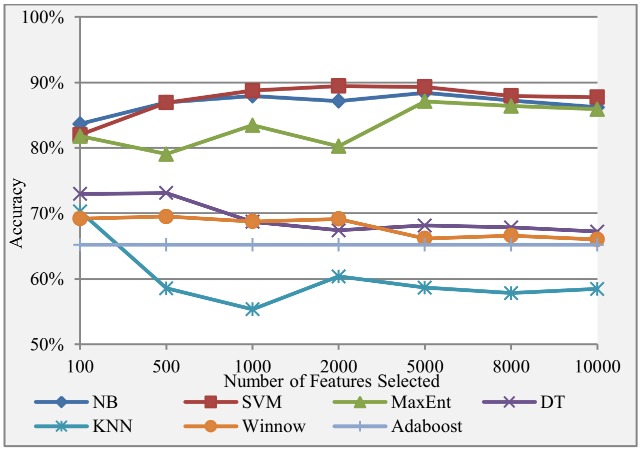

Figure 4. Performance of Machine Learning Techniques with Chi Squared Feature Selection.

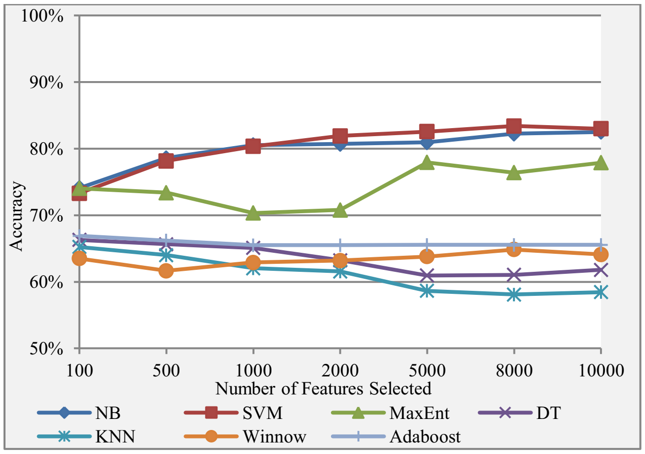

Figure 5. Performance of Machine Learning Techniques with Relief-F Feature Selection.

As such, among the machine learning classification methods, SVM produces the best accuracy most of the time when experimented with different number of selected features. Naïve Bayes classifier was the next best to SVM while NB and SVM outperformed almost all other machine learning classifiers. This observation indicates that SVM, NB and ME are all suitable for sentiment analysis but effective feature selection is critical for their performance.

While comparing NB with SVM, NB is a popular classification technique due to its simplicity but its strong conditional independence assumption clearly does not hold in tasks like sentiment analysis. The sentiment bearing features are inherently dependent on each other but NB based sentiment categorization still tends to perform surprisingly well as confirmed in several other studies [12, 21, 28, 31]. SVM has performed well as a non-linear, non-parametric classification technique, which has already showed good results in other sentiment analysis studies [1, 3, 4, 6, 17]. Another finding from the experiments was that, SVM was found abysmally slow, both in training and testing as compared to NB and ME. This paper has limited its scope to use classification accuracy as performance criteria so, execution time is not reported in the performance evaluation.

KNN is found to be highly sensitive to number of feature selected, when experimented with different feature selection methods. Adaboost and Winnow are found to give poor performance but gave stable result when executed with different feature selection methods. Adaboost requires documents to be represented by binary vectors, indicating presence or absence of the terms in the document [9]. As a consequence, this algorithms cannot take full advantage of the weighted term frequency representations and feature selection methods. From Figure 1-5, we can observe that when the number of features exceeds 2000, all learning methods produce desirable and reasonable performance. For example, using a feature set larger than 2000, the performance curves of SVM and NB remain nearly unchanged.

5. CONCLUSION AND REMARKS

This work has explored the applicability of feature selection methods on topical text classification for sentiment analysis. We have conducted a comparative study of feature selection methods for sentiment categorization on online movie reviews. The main contribution of this work is the performance investigation of different feature selection and machine learning methods in terms of accuracy. Results show that Gain Ratio performs the best among sentiment feature selection methods and SVM demonstrates the best performance for the sentiment classification task, while Naïve Bayes classifier gave better results when used with fewer features. Moreover, the experimental results indicate that number of features from 2000 to 8000 is sufficient for sentiment analysis. Our future efforts will targeted at investigating cross domain sentiment classification performance of these machine learning methods.

6. REFERENCES

[1] Annett, M., and Kondrak, G. A comparison of sentiment analysis techniques: Polarizing movie blogs. *Advances in Artificial Intelligence*, (2008), 5032, 25–35.

[2] Berger, A. L., Pietra, V. J. D., and Pietra, S. A. D. A maximum entropy approach to natural language processing. *Comput. Linguist*., 22, 1 (1996), 39-71.

[3] Boiy, E., Hens, P., Deschacht, K., and Moens, M. F. Automatic sentiment analysis of on-line text. In *Proceedings*

of the 11th International Conference on Electronic Publishing (Vienna, Austria) 2007.

[4] Chen, C., Ibekwe-SanJuan, F., SanJuan, E., and Weaver, C. Visual analysis of conflicting opinions. In *IEEE Symposium on Visual Analytics Science and Technology*, 2006, 59–66.

[5] Cui, H., Mittal, V., and Datar, M. Comparative experiments on sentiment classification for online product reviews. In *Proceedings of AAAI* (Boston, Massachusetts, July 16-20, 2006). 2006, 1265–1270.

[6] Dasgupta, S., and Ng, V. Topic-wise, sentiment-wise, or otherwise? identifying the hidden dimension for unsupervised text classification. In *Proceedings of the EMNLP'09* (Morristown, NJ, USA), ACL, 2009, 580–589.

[7] Dave, K., Lawrence, S., and Pennock, D. M. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international WWW conference* (Budapest, Hungary, May 20–24, 2003). 2003, 519–528.

[8] Eirinaki, M., Pisal, S., and Singh, J. Feature-based opinion mining and ranking. *Journal of Computer and System Sciences*, 78, 4 (2012), 1175-1184.

[9] Freund, Y., and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*. Springer-Verlag, 1995.

[10] Gamon, M. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the 20th international conference on Computational Linguistics* (Geneva, Switzerland). ACL, 2004.

[11] Joachims, T. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the ECML'98*, 1998, 137–142.

[12] Kang, H., Yoo, S. J., and Han, D. Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews. *Expert Systems with Applications*, 39, 5 (2012), 6000-6010.

[13] Kennedy, A., and Inkpen, D. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22(2) (2006), 110–125.

[14] Kononenko, I. Estimating attributes: analysis and extensions of RELIEF. In *Proceedings of the European conference on machine learning*. Springer-Verlag New York, Inc., 1994.

[15] Liu, B. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing*, 2010, 627-666.

[16] McCallum, A., and Nigam, K. A comparison of event models for Naive Bayes text classification. In *AAAI/ICML-98 workshop on learning for text categorization* (Menlo Park, CA) AAAI Press, 1998, 41–48.

[17] Paltoglou, G., and Thelwall, M. A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of the 48th Annual Meeting of the ACL*, 2010, 1386–1395.

[18] Pang, B., and Lee, L. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting of the ACL* (Barcelona, Spain, July 21–26, 2004). 2004, 271–278.

[19] Pang, B., and Lee, L. *Opinion mining and sentiment analysis*. Foundations and Trends in Information Retrieval, 2(1-2), (2008), 1-135.

[20] Pang, B., and Lee, L. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting of the ACL* (University of Michigan, USA, June 25–30, 2005). 2005, 115–124.

[21] Pang, B., Lee, L., and Vaithyanathan, S. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*. ACL, 2002.

[22] Quinlan, J. R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[23] Shimada, K., and Endo, T. Seeing several stars: A rating inference task for a document containing several evaluation criteria. In *Proceedings of the PAKDD*. Springer, LNCS volume 5012, 2008, 1006–1014.

[24] Tan, S., and Zhang, J. An empirical study of sentiment analysis for Chinese documents. *Expert Systems with Applications*, 34, 4 (2008), 2622-2629.

[25] Tang, H., Tan, S., and Cheng, X. A survey on sentiment detection of reviews. *Expert Systems with Applications*, 36, 7 (2009), 10760-10773.

[26] Wang, S., Li, D., Song, X., Wei, Y., and Li, H. A feature selection method based on improved fisher's discriminant ratio for text sentiment classification. *Expert Systems with Applications*, 38, 7 (2011), 8696-8702.

[27] Wilson, T., Wiebe, J., and Hoffmann, P. Recognizing contextual polarity: an exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(5) (2009), 399–433.

[28] Xia, R., Zong, C., and Li, S. Ensemble of feature sets and classification algorithms for sentiment classification. *Information Sciences*, 181, 6 (2011), 1138-1152.

[29] Yang, Y., and Lin, X. A re-examination of text categorization methods. In *Proceedings of the SIGIR'99*, 1999, 42–49.

[30] Yang, Y., and Pedersen, J. O. A comparative study on feature selection in text categorization. In *Proceedings of the ICML*, 1997, 412–420.

[31] Ye, Q., Zhang, Z., and Law, R. Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert Systems with Applications*, 36, 3 (2009), 6527–6535.

[32] Zhai, Z., Xu, H., Kang, B., and Jia, P. Exploiting effective features for Chinese sentiment classification. *Expert Systems with Applications*, 38, 8 (2011), 9139-9146.

[33] Zhang, T. Regularized winnow methods. *Advances in Neural Information Processing Systems*, 13 (2001), 703–709.

[34] Zhang, Z., Ye, Q., Zhang, Z., and Li, Y. Sentiment classification of Internet restaurant reviews written in Cantonese. *Expert Systems with Applications*, 38, 6 (2011), 7674-7682.