Some Investigations on Machine Learning Techniques for Automated Text Categorization

Bhagirath Prajapati
A. D. Patel Institute of Technology
New V.V. Nagar
Karamsad -388121 (India)

Sanjay Garg
Nirma University
Sarkhej-Gandhinagar Highway
Ahmedabad – 382481(India)

N C Chauhan
A. D. Patel Institute of Technology
New V.V. Nagar
Karamsad -388121 (India)

ABSTRACT
The automated categorization (classification) of texts into predefined categories is one of the widely explored fields of research in text mining. Now-a-days, availability of digital data is very high, and to manage them in predefined categories has become a challenging task. Machine learning technique is an approach by which we can train automated classifier to classify the documents with minimum human assistance. This paper discusses the Naïve Bayes, Rocchio, k-Nearest Neighborhood and Support Vector Machine methods within machine learning paradigm for automated text categorization of given documents in predefined categories.

Keywords
Machine learning, Text categorization.

1. INTRODUCTION
In this cyber age, availability of digital document is increased drastically. Accessing the documents in convenient way has become difficult task as number and size of documents growing day by day. One such task is Text Categorization (TC), which means to label natural language text in predefined categories.
Earlier Knowledge Engineering (KE) techniques were used for TC. KE is used in expert system which consists of manually defined logical rules of Disjunctive Normal Form (DNF) of type:

if (DNF formula) then (category);

Document can be classified under particular category only if it satisfies the rule. The drawback of is this approach is the knowledge acquisition bottleneck. It is process in which expert person have to form DNF for new category. In last decade, the Machine Learning (ML) approach has gained popularity. In this approach, a general inductive process (learner) automatically builds a classifier. Learner automatically classifies document in predefined categories.
Automated text categorization of documents in predefined categories is becoming popular in this digital age. Because now-a-days availability of digital documents increase dramatically, it becomes necessary to investigate and develop novel techniques for automated text categorization.
Automated text categorization is applicable in document organization. In document organization documents has to be categorized in appropriate category. For example, in news paper agency the incoming advertisement has to be classified in one of the category like real estate, car for sale, office on rent. If such task is done manually than it would take lots of time. Automated systems are required that can accept the advertisement as input and categorize it to one of predefined categories. Let us take one more example where classifying the dynamic collection of text is to be done. Consider the example of e-mail filtering, where the computerized system is trained on "spam" mails to filter it out from non-spam mails [2].
Machine learning is an area of artificial intelligence. Machine learning deals with the study of methods for making computers learn like humans. Automated techniques from AI and machine learning have been developed to handle many problems of pattern recognition/categorization. One such task is text categorization of documents. Traditionally text categorization task is being carried out by KE techniques. In KE techniques human assistant is needed for forming decision rules for categorizing individual categories. So the idea is to explore the application of machine learning techniques for automated text categorization [2, 3], which can be free from human interactions. Automated text categorization with machine learning gained a prominent status in the information systems field. In this technique, a learner is implemented which automatically learn from previously classified documents.
As discussed from this applications and importance of automated text categorization system encourages implementing computerized system which can classify incoming documents into appropriate predefined category. In implementing automated text categorization system the technique of machine learning algorithm which can be trained for some labeled documents and able to classify the incoming unlabeled documents into its appropriate category with high accuracy is to be explored.
There are many machine learning algorithms available to build a learner for text categorization system. So it is interesting to implement few popular techniques of classification of text and to perform a comparative analysis in term of accuracy for such techniques. This paper deals with following objects. The first is to explore basic preprocessing steps for text categorization. It also presents the study of some machine learning techniques for text classification. The paper also focuses on investigation of performance issues in text categorization.

2. INFORMATION RETRIEVAL
Information retrieval (IR) is a process of searching material (usually documents) of an unstructured nature (usually text) that satisfies the user's need from large collections [1]. These documents may be stored on single computer or available on web.

2.1 Automatic Text Classification

Information retrieval system needs information to be retrieved, which is usually stored in the form of documents. Documents are presented in the form of some standard representation. The starting point of the text analysis process may be the complete document text, an abstract, the title only, or perhaps a list of words only. From this, the process must produce a document representative in a form which the computer can handle. According to Luhn's Idea [2]: 'the frequency of word occurrence in an article furnishes a useful measurement of word significance'. This is shown in Fig. 1.

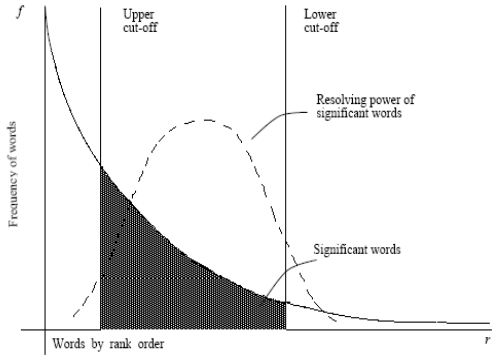

Fig.1 Lun's idea about frequency of terms [2]

Prior to computation of frequency of terms, the document is first preprocessed. In preprocessing generally three things are carried out.

- Removal of high frequency words
- Suffix stripping
- Detecting equivalent stems

The removal of high frequency words, 'stop' words or 'fluff' words is one way of implementing Luhn's upper cut-off. This is normally done by comparing the input text with a 'stop list' of words which are to be removed. The second stage, suffix stripping is difficult, the standard approach is to have list of suffix and remove only the longest one [2]. Many times context free removal leads to error. For example, if UAL is to be removed from FACTUAL than it may be fine but if it is to be done on EQUAL than obviously the meaning of the word is lost. Such way stemming is done.

3. MACHINE LEARNING METHODS FOR TEXT CATEGORIZATION

Machine learning is the study of methods by which computers learn and reacts like human. There are many tasks which are considered to be difficult or impossible. These tasks can be divided into four general categories as mentioned in [4]. Text categorization (TC) is the problem of assigning predefined categories to free text documents. A growing number of statistical learning methods have been applied to this problem in recent years.

3.1 Steps Prior to Inductive Process

The following steps are required to be performed before applying the inductive process.

Step1: The classification problem is supervised learning in terminology of ML. ML technique relies on training set, and test set. Training set is participating in inductive process to build a classifier, while the test set do not take participate in this inductive process. So it is required to have initial corpus which can be divided into training set and test set [3].

Step2: Before classifying text directly by inductive classifier prior to inductive process, indexing procedure for a document d_j is needed. *Bag of Words* approach [3] is used to represent a document for indexing. In this approach weight ω_k for each term t_k document d_j is calculated. Most of the time standard *tf-idf* function is used [5] as,

$$tf\text{-}idf(t_k, d_j) = \#(t_k, d_j) \cdot \log T_r / \sim T_r \quad \ldots (1)$$

where *#(tk, dj)* denotes the number of times t_k occurs in d_j, and $\sim T_r$ denotes the document frequency of term t_k, that is, the number of documents in T_r in which t_k occurs. Before indexing, *stemming* is performed on words [6]. Stemming process groups words that share the same morphological roots and also it reduce dimensionality and term space.

Step3: For TC high dimensionality of term space is not proper for many sophisticated algorithms (e.g. LLSF [8]). Hence, before classification, dimensionality reduction (DR) is applied. There are two different methods for dimensionality reduction [8]. The first DR method is by term selection: reduced terms T' is subset of original terms T. The second DR method is by term extraction: the terms in T' are not of the same type of the terms T. T' are generated from original terms by way of transformation.

3.2 Inductive Process

In the inductive process, learning algorithm is applied. Here, four such machine learning algorithms have been investigated.

3.2.1 Naïve Bayes Text Classifier

In Naïve Bayes (NB) classification [7] the probability a document d_j being in class c_i is computed as,

$$P\left(\frac{c_i}{d_j}\right) \propto P(c_i) \prod_{i \le k \le n_{d_j}} p\left(\frac{t_k}{c_i}\right) \quad \ldots (2)$$

Where $p(t_k / c_i)$ is the conditional probability of term t_k occurring in document class *c*. *P(c)* is prior probability of a document occurring in class $c_i \langle t_1, t_2, \ldots t_{n_d} \rangle$ are tokens in d_j that are part of the vocabulary we used for classification and n_d is the number of such tokens in d_j [7]. In text classification, the goal is to find the best class for the given document, in NB the most likely class is known as *maximum a posteriori* (MAP) class:

$$\therefore c_{map} = \arg_{c_i \in C} \max P(c_i) \prod_{i \le k \le n_{d_j}} P\left(\frac{t_k}{c_i}\right) \quad \ldots (3)$$

In NB, the estimation of the parameters $P(c_i)$ and $P(t_k/c_i)$ is important. For priors this estimate is $P(c_i) = \frac{N_{c_i}}{N}$ where, Nc_i is the number of documents in class *c*, and *N* is the total number of documents. And $P(t_k/c_i)$, can be given as,

$$P\left(\frac{t_k}{c_i}\right) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}$$

where $T_{ct'}$ is the number of occurrences of t_k in training documents for class c_i.

3.2.2 Rocchio Classifier

The Rocchio method is a linear classifier [3]. In this method, text is presented by vector $\vec{c} = \langle w_{1i}, w_{2i}, \ldots w_{ri} \rangle$ .Given a training dataset T_r, it directly computes a classifier for category c_i by means of the formula:

$$w_{ki} = \beta . \sum_{d_j \in POS_i} \frac{w_{kj}}{|POS_i|} - \gamma . \sum_{d_j \in NEG_i} \frac{w_{kj}}{|NEG_i|} \quad ...(4)$$

where w_{ki} is the weight of the term t_k in the document d_j,

$$POS_i = \{d_j \in T_r \mid \breve{\phi}(d_j, c_i) = T\},$$

and

$$NEG_i = \{d_j \in T_r \mid \breve{\phi}(d_j, c_i) = F\},$$

$\breve{\phi}(d_j, c_i) = T$ (or F) means document d_j belonging to category c_i. In Equation 4, β and γ are two control parameters used for setting the relative importance of positive and negative instances. The profile of c_i is the centroid of its positive training examples. A classifier built by means of the Rocchio method rewards the closeness of a test document to the centroid of the positive training instances, and its distance from the centroid of the negative training instances

3.2.3 k-Nearest Neighborhood Classifier

K-Nearest Neighborhood (*k-NN*) is similarity based learning algorithm. It has wide application including TC. Given an arbitrary input document, the system ranks its nearest neighbors among the training documents, and uses the categories of the *k* top-ranking neighbors to predict the categories of the input document. It may happen that several neighbors share same category. Amongst this *k* nearest neighbors chosen, then the per-neighbor weights of that category are added together, and the resulting weighted sum is used as the likelihood score of candidate categories. The score is compared with threshold value for document d_j , and if the weighted sum is greater than threshold to indicate the category c_i is applicable for document d_j than value *T*(*true*) is assigned else value *F*(*false*) is assigned to document d_j under the category c_i [6].

3.2.4 Support Vector Machine Classifier

Support Vector Machine (SVM) [9] attempts to find, among all the surfaces δ_1, δ_2, … in $|T|$dimensional space that separate the positive from the negative training examples (*decision surfaces*), the δ_i that separates the positives from the negatives by the widest possible margin, that is, such that the separation property is invariant with respect to the widest possible translation of δ_i.

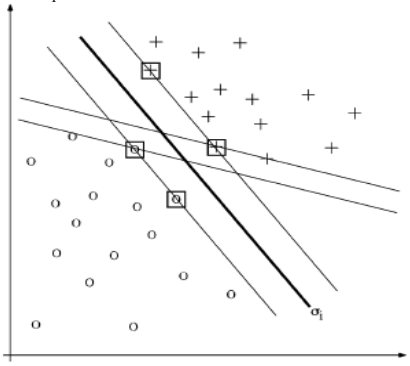

Fig 2. Learning support vector classifiers [9].

This idea is best understood in the case in which the positives and the negatives are linearly separable, in which case the decision surfaces are (|*T*|-1)-Hyperplanes. In the two-dimensional case of Figure 2, various lines may be chosen as decision surfaces. The SVM method chooses the middle element from the "widest" set of parallel lines, that is, from the set in which the maximum distance between two elements in the set is highest. It is noteworthy that this "best" decision surface is determined by only a small set of training examples, called the *support vectors.*

4. Evaluation of Classifier

In text categorization system, the evaluation is done by standard methods of information retrieval. The evaluation measures revolve around the notion of relevant and non-relevant documents. The start point is to compare the matches between human-assigned key words and computer assigned ones. We can summarize four possible situations in the following contingency Table 1 [7].

Table 1. Contingency Table

Category Expert Judgments C_i		**Expert Judgments**	
		YES	**NO**
Classifier Judgments	YES	TP_i	FP_i
	NO	FN_i	TN_i

6.1 Precision (P)

It is fraction of retrieved documents that are relevant [10]. In other words precision, measures how many of the results returned are actually relevant as per Equation (5).

$$precision = \frac{relevant\ items\ retrieved}{retrieved\ items} \quad ...(5)$$

From contingency table, precision can be calculated as per Equation (6).

$$precision = \frac{TP_i}{TP_i + FP_i} \quad ...(6)$$

6.2 Recall (R)

It is the fraction of relevant documents that are retrieved [10]. In other words recall measures how large a fraction of the expected results is actually found. It is found as shown in Equation (7).

$$recall = \frac{relevant\ items\ retrieved}{total\ relevant\ items} \quad ...(7)$$

From contingency table, recall can be calculated as per Equation (8).

$$recall = \frac{TP_i}{TP_i + FN_i} \quad ...(8)$$

6.3 Accuracy

It is fraction of its classification that is correct as per Equation (9).

$$\text{Accuracy} = \frac{TP_i + TN_i}{TP_i + FP_i + FN_i + TN_i} \quad \ldots(9)$$

7. Experimental Results

7.1 Experimental Set-Up

For automated text categorization many datasets are available. One of such data set is 20-newsgroup [11]. This dataset is available in raw text format which has to be converted to the standard representation of vectors. The dataset is divided into two different sets, one is training and other is test set. The classifier is first trained with training set and the test set is applied to check the accuracy of the classifier.

7.2 The Dataset

In this experiment 20-newsgropup data set [11] is used. It is collection of electronics text documents. The documents are already categorized in predefined categories in the training and test document set. In this experiment, two such categories of documents – Religion and Politics have been used.

In all of the experiments, five randomly chosen datasets from the above mentioned dataset have been used to test and compare the performance of different classifiers. Each dataset consists of two file, for training and testing. Each file represents two classes of documents, 200 documents are of religion and 200 documents are belongs to political class.

7.3 *k*-NN Classification

In this experiment, five different values of k – 3, 7, 11, 15 and 17, were considered. The classification results are presented in Table 2.

Table 2. Accuracy of *k*-NN Classification for different values of *k*

Data Set	Accuracy (%)				
	k-NN (*k*=3)	*k*-NN (*k*=7)	*k*-NN (*k*=11)	*k*-NN (*k*=15)	*k*-NN (*k*=17)
Data-0	80.25	85.25	87	86.25	**87**
Data-1	82.5	89	92.25	94	**94.75**
Data-2	84.5	85.5	93	94.5	**95.50**
Data-3	81	89.75	96	97.75	**98.25**
Data-4	80.5	84.5	87.75	89.25	**90.75**

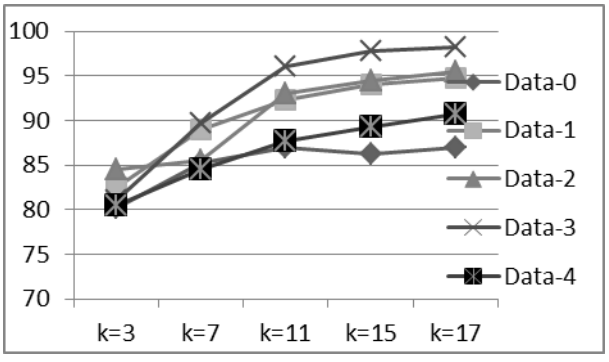

Fig 3: Accuracy for different values of k

7.4 Comparison of Different Classifiers

In these experiment four different classifiers, namely, Rocchio, *k*-NN, Naïve Bayes and SVM were used for evaluation. The comparison of accuracy for each classifier on 5 different datasets are given in Table 3 and also shown in Fig. 4.

Table 3. Comparison of accuracy for different classifiers

Data Set	Accuracy (%)			
	Rocchio	*k*-NN (*k*=17)	Naïve Bayes	SVM
Data-0	86	87	85	**94.75**
Data-1	71.75	**94.75**	87.25	93.25
Data-2	83	95.50	83	**96.5**
Data-3	59	**98.25**	82.25	91.5
Data-4	88.50	90.75	88	**92.5**

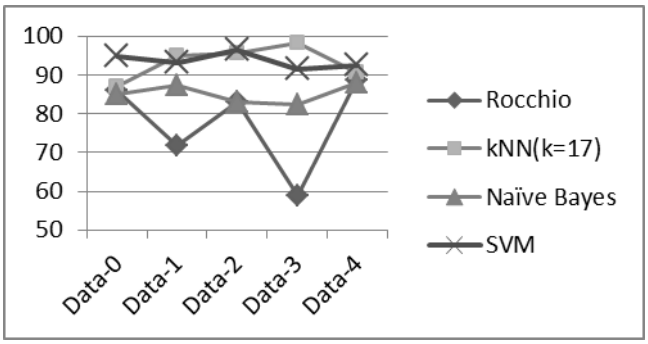

Fig 4: Accuracy for different classifiers

8. Conclusion

Automated text categorization has been found to be major research area due to increasing availability digital documents. The process of automated text categorization requires documents in some standardized format. This preprocessing of converting the documents in vector form of M x N term document matrix was performed successfully. Major contributions of this work is summarize as below.

- Preparation of the dataset of training and test documents is performed from available document corpus of 20 news group. Indexing with stop word removal and stemming is done and documents are converted into vector form.
- A probabilistic classifier, Naïve Bayes has been investigated for TC which predicts the class based on probability of the term.
- Implementation of linear classifier, Rocchio been carried out for TC which determines the category of document by means of linear separator vector.
- Implementation of sample based classifier, K-Nearest Neighbor (*k*-NN) has been carried out. Already labeled documents determine the category incoming document in this classifier. In *k*-NN classifier *k* indicates number of neighbors to be considered for taking classification decision.
- Performance measure of *k*-NN for increasing value of k is experimented. From experiment it becomes clear that as value of *k* increases the accuracy of classifier increases.
- Implementation of support vector machine is been carried out for the problem of TC.

- Comparative performance of classifier Naïve Bayes, Rocchio, *k*-NN, and support vector machine is shown. From performance criterion like accuracy, it becomes clear that SVM and *k*-NN classifiers performs better compared to Naive Bayes, and Rocchio classifier techniques.

9. REFERENCES

[1] Manning, C. D., Raghavan, P., Chütze, H. 2009. An Introduction to information retrieval, Chapter 1: Boolean retrieval, page 1, Cambridge University Press.

[2] Rijsbergen, C. J. V. 1979. Information retrieval: Chapter 2: Automatic Text Analysis, Butterworth-Heinemann, 2nd edition.

[3] Sebastian, F., Ricerche, C. N. 2002. "Machine learning in automated text classification", ACM Computing Surveys, Vol. 34, No.1, pp. 1-47.

[4] Nilsson, N. J. 1996. Introduction to machine learning, Chap 01: Preliminaries, Draft of Incomplete.

[5] Salton, G., Buckley, C. 1988.Term-weighting approaches in automatic text retrieval. Information Processing and Management, 24(5), pages. 513–523.

[6] Guo, G., Wang, H., Bell, D., Bi, Y., and Greer, K. 2006. "Using k-NN model-based approach for automatic text categorization", Soft Computing-A Fusion of Foundations, Methodologies and Applications.

[7] Manning, C., Raghvan, P., and Schutze, H. 2008. "Text classification and Naïve Bayes", Chapter in Introduction to Information Retrieval, Cambridge University Press.

[8] Yang, Y. 1994. "Expert network: effective and efficient learning from human decisions in text categorization and retrieval", In Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval, Dublin, Ireland, pages. 13–22.

[9] Joachims, T. 1999. "Transductive inference for text classification using support vector machines", ICML-99, Pages 200–209.

[10] Yang, Y., Liu, X. 1999. "A re-examination of text categorization methods", SIGIR-99, Page 42–49.

[11] Vang, K.: 20 news group dataset, http://people.csail.mit.edu./Jrennie/20newsgroup.