IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

A Different Type of Feature Selection Methods for Text Categorization on Imbalanced Data

Senthil Kumar B[1], Bhavitha Varma E[2]

Assistant Professor, Department of Computer Science, Sree Narayana Guru College, Coimbatore, India [1]

M.Phil Scholar, Department of Computer Science, Sree Narayana Guru College, Coimbatore, India [2]

Abstract: Text categorization is an important and well-studied area of pattern recognition, with a variety of modern applications. Effective spam email filtering systems, automated document organization and management, and improved information retrieval systems all benefit from techniques within this field. The problem of feature selection, or choosing the most relevant features out of what can be an incredibly large set of data, is particularly important for accurate text categorization. The proposed system (i) use well known pre-processing method porter and Lancaster for train the dataset. (ii) A number of feature selection metrics have been explored in text categorization, among which information gain (IG), chi-square (CHI), Mutual information (MI), Ng-Goh-Low (NGL), Galavotti-Sebastiani-Simi (GSS), Relevancy Score (RS), Multi-Sets of Features (MSF) Document frequency (DF) and odds ratios (OR) are considered most effective. Pruning techniques are also proposed using ignore the feature based on TF and DF to further reduce the set of possible features (typically words) within a document prior to applying a method of feature selection. (iii) Finally classify the selected feature based on two algorithm KNN and Navie bayes. Two benchmark collections were chosen as the testbeds: Reuters-21578 and small portion of Reuters Corpus Version 1 (RCV1). The two classifiers and both data collections, and that a further increase in performance is obtain by combining uncorrelated and high-performing feature selection methods.

Keywords: Locally Weighted Spectral Cluster, matrix, Local Scaling, Estimating Weight based Clusters

I. INTRODUCTION

The purpose of text classification is to use the contents of a text or document to assign it to one or more categories. It has applications in document organization and management, information retrieval, and certain machine learning algorithm. More effective spam email filtering systems, improved web search results, and better translations between languages can result from improved text classification techniques.
Feature selection has been applied to text categorization in order to improve its scalability, efficiency and accuracy. Since each document in the collection can belong to multiple categories, the classification problem is usually split into multiple binary classification problems with respect to each category. Accordingly, features are selected locally per category, e.g. local feature selection.
One choice in the feature selection policy is whether to rule out all negative features.

Some argue that classifiers built from positive features only may be more transferable to new situations where the background class varies. Others believe that negative features are numerous, given the imbalanced data set, and quite valuable in practical experience. Their experiments show that when deprived of negative features, the performance of all feature selection metrics degrades, which indicates negative features are essential to high quality classification. We think that negative features are useful because their presence in a document highly indicates its non-relevance. Therefore, they help to confidently reject non-relevant documents. Two principles should keep in mind: a) most (digital) documents can be reduced to their text content, and b) similarity between documents is utterly important. One traditional technique of trying to tackle these problems is the vector space model. Here, every set of documents (or rather every collection) is described by all terms occurring in it (i.e. a collection of e-mails, an inbox for example). Every single document is presented by a set of terms occurring in the particular document. Hence, every document is represented by a vector in which each position represents one of the overall terms in the collection.

This can range from a boolean vector (if a document contains a term the value at the given position is 1 and 0 otherwise) to more sophisticated term weighting approaches. Once a vectorial representation is obtained, similarity computation can be performed on it. One area within text mining is text categorisation. Herein, the main task is to automatically assign class labels to unlabelled documents (e-mail messages for example are labelled as categorised in two classes, either ham or spam). In a typical spam filter scenario, for instance, a training corpus is available (a certain number of both spam and ham messages) and new unlabelled messages are assigned one of the classes based on the data found in the training corpus. For instance, an e-mail message about might rather be categorised as spam based on the evidence found in the training collection (most messages containing the term will be labelled as spam). The actual process of deciding how to label a message is done via machine learning algorithms. The main idea is to implement a range of these feature selection techniques with the goal of a thorough performance evaluation on different test collections. Comparisons reported in the literature are often somewhat difficult for articles working with feature selection often use different underlying techniques or parameters.

II. RELATED WORK

These two different feature models are both widely used in TC for classification as well as for feature selection. Under the probabilistic framework of naive Bayes, the Binary-valued feature model is used in Bernouli naive Bayes (BNB), and the Real-valued feature model is used in multinomial naive Bayes (MNB) or Poisson naïve Bayes (PNB). For classification, empirical studies have shown that the Real-valued feature model offers better performance than the Binary-valued feature model. Interestingly, at the stage of feature selection, the Binary-valued feature model is more commonly used than the Real-valued one. We will present the feature selection approach in detail using these two feature models in the following section.
For classification, naive Bayes gains popularity due to its efficiency and simplicity. Empirical studies have shown that naive Bayes could provide competitive performance compared with the state-of-the-art discriminative classifiers, such as support vector machine, nearest neighbours-based methods, etc. The naive Bayes is a model-based classification method with the "naive" assumption of independent features, and three distribution models are usually applied, including Bernoulli model, multinomial model and Poisson model, which result in the classifiers of Bernoulli naive Bayes, multinomial naive Bayes and Poisson naive Bayes, respectively. Previous studies on real-life benchmarks have shown that the MNB usually performs better than the BNB at large vocabulary size and the PNB is equivalent to the MNB when the document length and document class are assumed to be independent. For this reason, we commonly refer the naive Bayes to the MNB classifier.
Many feature selection methods have been proposed in general machine learning fields, such as regression, classification, and clustering. Some of those methods can be also used for text categorization which can be considered as a multi-class classification problem. Relevance of features is a major concern for designing feature selection methods. For example, several well-recognized feature selection methods have been developed considering the entropic relevance, such as document frequency, information gain], mutual information statistic, etc. In a comparative analysis of these methods is presented. In, an extensive empirical study is performed using these feature selection methods for text categorization. The empirical results show that feature selection methods can effectively reduce the computation of learning and speed up the learning process with little loss of discriminative performance. To find a suitable feature subset for a learning algorithm, several feature selection methods are usually needed to test and compare. It is difficult to select the optimal feature subset in a theoretical way.
The existng authors used an in-house collected corpus from online Arabic newspaper archives, including Al-Jazeera, Al-Nahar, Al-Hayat, Al-Ahram, and Al-Dostor. The collected corpus consists of 1445 documents. These documents consist of nine categories, the authors did some Pre-processing for the dataset such as remove digits and punctuation marks, all the non-Arabic texts were filtered, remove the Arabic function words (stop words) and other. In the result showing that CHI, NGL and GSS performed most effective with SVMs for Arabic TC tasks, but OR and MI performed terribly. In [3] the authors talked about three contributions: (i) showing successful classification of Arabic documents, (ii) make their database available to other researchers, (iii) find a better performance between Binary PSO and K-nearest neighbour using feature selection methods. In] the authors presented BPSO - KNN as a feature selection method and applied this method on three Arabic text dataset. The authors used three classification algorithms which are SVM, Naïve Bayes andC4.5 decision tree learning.
Common general techniques for text classification include both unsupervised and super. vised pattern classification methods. Some common approaches use clustering instead of simple feature selection linear discriminant methods], neural networks, and support vector machines. Some models attempt to use linguistic information in the classification process, such as labelled samples from the WordNet data set provides an excellent survey of text classification methods, and provides a summary and empirical study of 12 feature selection metrics.
Content analysis is an approach that involves a deeper understanding of the semantics of text and other media items (especially pictures), by using linguistic analysis, machine learning, and image processing components. Currently content based analysis approach becomes a new trend for web filtering research. For instance, Lee et al. use Kohonen's Self-Organizing Maps (KSOM) and Fuzzy Adaptive Resonance Theory (Fuzzy ART) for their online pornography document classification, show a machine learning-based system for pornography web page classification by combines textual and structural content-based analysis, and Zhou et al. using link and content analysis for detecting US domestic extremist groups on the web. However practical implementation on each solution still remain as an issue. The extract the features of web page by simply select certain numbers of terms as their features extraction algorithm. The problem issues may rise when the web pages contain the terminologies that not including in the system terms dictionary. The identify the terrorist web content via text analysis without concerning the similarity content, thus the system design may be facing problem when identifying similar subject such as terrorist and military.

From the textual content analysis point of view, natural language is redundant in the sense that many different words are sharing a similar meaning. As for computer, it is hard to understand the meaning of natural language without some proper ways]. Term weighting scheme is a statistical measure used to evaluate the important of a word in the collection of documents. In other words, there will be set of numeric number that obtained through term weighting scheme. These sets of numbers will be understood by computer machine that will be used for further analysis and document classification.

III.PROPOSED APPROACH

The imbalanced data problem occurs when the training examples are unevenly distributed among different classes. In case of binary classification, the number of examples in one class is significantly greater than that of the other. Attempts have been made to deal with this problem in diverse domains such as fraud detection, in sight helicopter gearbox fault monitoring, and text categorization.

A. Dataset

Reuters Corpus Volume 1 (RCV1) data set which contains over 800,000 documents and the data dimension is about 500,000. We choose the data samples with the highest four topic codes (CCAT, ECAT, GCAT, and MCAT) in the “Topic Codes” hierarchy, which contains 789,670 documents. Then we split them into 5 equal-sized subsets, and each time 4 of them are used as the training set and the remaining ones are left as the test set. The experimental results reported in this paper are the average of the runs. Moreover, we use this dataset as a single label problem, i.e. we only keep the first label if a sample is multi-labelled.

B. Pre-processing

Removal of Stop Words: In most of the applications, it is practical to remove words which appear too often (in every or almost every document) and thus support no information for the task. Good examples for this kind of words are prepositions, articles and verbs like” be” and” go”. If the box” Apply stop word removal” is checked, all the words in the file” swl.txt” are considered as stop words and will not be loaded. This file contains currently the 100 most used words in the English language which on average account for a half of all reading in English. If the box” Apply stop word removal” is unchecked, the stop word removal algorithm will be disabled when the corpus is loaded.

Stemming: Stemming or lemmatisation is a technique for the reduction of words into their root. Many words in the English language can be reduced to their base form or stem e.g. agreed, agreeing, disagree, agreement and disagreement belong to agree. Furthermore, are names transformed into the stem by removing the” ’s”. The variation” Peter’s” in a sentence is reduced to” Peter” during the stemming process. The result of the removal may lead to an incorrect root. However, these stems do not have to be a problem for the stemming process, if these words are not used for human interaction. The stem is still useful, because all other inflections of the root are transformed into the same stem. Case sensitive systems could have problems when making a comparison between a word in capital letters and another with the same meaning in lower case. Following a selection of suffixes and prefixes for removal during stemming

- **suffixes:** ly, ness, ion, ize, ant, ent , ic, al , ical, able, ance, ary, ate, ce, y, dom , ed, ee, eer, ence, ency, ery, ess, ful, hood, ible, icity, ify, ing, ish, ism, ist, istic, ity, ive, less, let, like, ment, ory, ty, ship, some, ure
- **prefixes:** anti, bi, co, contra, counter, de, di, dis, en, extra, in, inter, intra, micro, mid, mini, multi, non, over, para, poly, post, pre, pro, re, semi, sub, super, supra, sur, trans, tri, ultra, un.

Porter Stemming: The idea of this algorithm is the removal of all pre- and suffixes to get the root of a word. The main field of application for the Porter Stemmer is languages with simple inflections, such as English. The algorithm is favoured and often used because of the simplicity and the small number of rules. Following an explanation of the algorithm, based on the publication of Martin F. Porter. The algorithm makes a distinction between consonants and vowels in a word. Therefore, the selection of the applying rules during the stemming process is based on the sequence of consonants and vowels.
A word is represented by the form **[C]VCVC ... [V]**
Where the notation of a sequence of VC is written as (VC){m}, with VC repeated m times. An example for a repetition with m = 0 is sea, for m = 1 is cat, for m = 2 is garden and so on. The further processing of the suffix stripping is decided by several conditions. One of the conditions was mentioned in the sentences before, the repletion of VC in a word.
The other conditions for the Porter Stemming are:
- *S - the stem ends with S (and similarly for the other letters).
- *v* - the stem contains a vowel.
- *d - the stem ends with a double consonant (e.g. -TT, -SS).
- *o - the stem ends CVC, where the second c is not W, X or Y (e.g. -WIL, -HOP).

Furthermore, combinations of these conditions are possible (using and, or and not). Following, the rules with some examples, divided into 5 steps. Only the application of one rule for a step is allowed. This rule has to remove the longest matching suffix.

Lancaster Stemming: Stemming is a well-known technique for information retrieval. The use of stems for searching has the advantage of increasing recall by retrieving terms that have the same roots but different endings. A major disadvantage of stemming is a decrease of precision as compared to the use of expanded terms. When searching with stems, it is not uncommon to retrieve many irrelevant terms that have similar roots but which are not related to the object of the search. Several commonly-used stemming programs and algorithms were evaluated to try to select a stemmer suitable for information retrieval of large databases. The flexibility of being able to specify a new set of rules without extensive programming changes made the Paice/Husk stemmer more attractive than the Porter stemmer.

C. Feature Selection Methods

Odds Ratio: Odds Ratio compares the odds of a feature occurring in one category with the odds for it occurring in another category. It gives a positive score to features that occur more often in one category than in the other, and a negative score if it occurs more in the other. A score of zero means the odds for a feature to occur in one category is exactly the same as the odds for it to occur in the other, since ln (1) = 0. The original Odds Ratio algorithm for binary categorization:

$$\mathbf{OR\ (F,\ C_k) = ln}\frac{P(F|C_k)(1-P(F|\overline{C_k}))}{P(F|\overline{C_k})(1-P(F|C_k))} = \mathbf{ln}\frac{\left(\frac{N_{F,C_k}}{NC_k}\right)\left(1-\frac{N_{F,\overline{C_k}}}{N\overline{C_k}}\right)}{\left(\frac{N_{F,\overline{C_k}}}{N\overline{C_k}}\right)\left(1-\frac{N_{F,C_k}}{NC_k}\right)}$$

$$\mathbf{P\ (F|C_k)} = \frac{N_{F,C_k}}{N_{C_k}}$$

Let P (t|c) be the probability of a randomly chosen word being t, given that the document it was chosen from belongs to a class c. Then odds (t|c) is defined as P (t|c)/ [1–P (t|c)] and the Odds Ratio equals to,
OR (t) = ln [odds (t|c+) / odds (t|c–)]
Obviously, this scoring measure favors features that are representative of positive examples. As a result, a feature that occurs very few times in positive documents but never in negative documents will get a relatively high score. Thus, many features that are rare among the positive documents will be ranked at the top of the feature list. Odds Ratio is known to work well with the Naïve Bayes learning algorithm.

Information Gain: Here both class membership and the presence/absence of a particular term are seen as random variables, and one computes how much information about the class membership is gained by knowing the presence/absence statistics (as is used in decision tree induction. Indeed, if the class membership is interpreted as a random variable C with two values, positive and negative, and a word is likewise seen as a random variable T with two values, present and absent, then using the information-theoretic definition of mutual information we may define Information Gain as:

IG(t) = H(C) – H(C|T) = Στ,c P(C=c,T=τ) ln[P(C=c,T=τ)/P(C=c)P(T=τ)]
Here, τ ranges over {present, absent} and c ranges over {c+, c–}. As pointed out above, this is the amount of information about C (the class label) gained by knowing T (the presence or absence of a given word).

Document Frequency (DF) Thresholding: One of the simplest methods of vocabulary reduction, and hence vector dimensionality reduction, is the Document Frequency Thresholding,
DF (F) = N_F
The number of documents containing a feature in the training set is counted. This is done for every feature in the training set, before removing all features with a document frequency less than some specified threshold and features with a frequency higher than some other threshold. Alternatively, the document frequency can be used as any other feature selection method where it creates a ranked list, and returns the highest ranked features.

Mutual Information: Mutual Information can be proven equal to Information Gain for binary problems. For mutli-class problems (with global feature lists) like we present in this report however, the two are not equal (although rather similar). Thus we present Mutual Information with its own equation as a separate feature selection algorithm here.

$$\text{MI (F, C}_k) = \sum_{\upsilon f \in \{1,0\}} \sum_{\upsilon f \in \{1,0\}} \text{P (F} = \upsilon_f, C_k = \upsilon_{C_k}) \ \ln \frac{P(F = \upsilon_f, C_k = \upsilon_{C_k}}{P(F = \upsilon_f)P(C_k = \upsilon_{C_k})}$$

Where F is the discrete random variable `feature' that takes the value ʋF = f1; 0g (feature F occurs in document or not), Ck is the discrete random variable `category' that takes the values ʋCk = f1; 0g (document belongs to category Ck or not). The probabilities can be estimated by using the various document counts from the training set.

$$\text{MI (F, C}_k) = \frac{N_{F,C_k}}{N}\ln\frac{NN_{F,C_k}}{N_F N_{C_k}} + \frac{N_{F,\overline{C_k}}}{N}\ln\frac{NN_{F,\overline{C_K}}}{N_F N_{\overline{C_k}}} + \frac{N_{\overline{F},C_k}}{N}\ln\frac{NN_{\overline{F},C_k}}{N_{\overline{F}} N_{C_k}} + \frac{N_{\overline{F},\overline{C_k}}}{N}\ln\frac{NN_{\overline{F},\overline{C_k}}}{N_{\overline{F}} N_{\overline{C_k}}}$$

Then the values can be weighted and summarized to create a global ranked list of features:

$$\text{MI (F)} = \sum_{k=1}^{|C|} \frac{N_{C_k}}{N} \text{MI (F, C}_k)$$

Chi Square (Chi): Feature Selection by X2 testing is based on Pearson’s X 2 (chi square) tests. The X2 test is often used to test the independence of two variables. The null-hypothesis is that the two variables are completely independent of each other. The higher value of the X2 test, the closer relationship the variables have. In feature selection, the X2 test measures the independence of a feature and a category. The null-hypothesis here is that the feature and category are completely independent, i.e. that the feature is useless for categorizing documents. The higher X2 value for a (feature, category) pair, the less independent they are. Hence, the features with the highest X2 values for a category should perform best for categorizing documents.

$$X^2(F, C_k) = \frac{N\ X\left(\left(N_{F,C_k}\ X\ N_{\overline{F},\overline{C_k}}\right) - \left(N_{F,\overline{C_k}}\ X\ N_{\overline{F},C_k}\right)\right)^2}{N_F\ X\ N_{\overline{F}}\ \ X\ N_{C_k}\ X\ N_{\overline{C_k}}}$$

NGL Coefficient: The NGL coefficient presented is a variant of the Chi square metric. The name it `NGL coefficient' after the last names of the inventors Ng, Goh, and Low. The NGL coefficient looks only for evidence of positive class membership, while the chi square metric also selects evidence of negative class membership. Hence, it is called a `one-sided' chi square metric. In their experiments, it performed better than chi square. It better than Odds Ratio and Mutual Information on some feature set sizes, and worse on other.

$$\boldsymbol{NGL(F, C_k) = \frac{\sqrt{N}\left(N_{F,C_k}N_{\overline{F},\overline{C_k}} - N_{F,\overline{C_k}}N_{\overline{F},C_k}\right)}{\sqrt{N_F N_{\overline{F}} N_{C_k} N_{\overline{C_k}}}}}$$

GSS Coefficient: The GSS coefficient was originally presented as a `simplified chi square function'. We follow and name it GSS after the names on the inventors Galavotti, Sebastian, and Simi.

$$\boldsymbol{GSS(F, C_k) = N_{F,C_k}N_{\overline{F},\overline{C_k}} - N_{F,\overline{C_k}}N_{\overline{F},C_k}}$$

The experiments showed far better results when using max as a globalizing strategy rather than average, hence we follow them on that:

$$\boldsymbol{GSS(F) = \max_{k=1} |C|\, GSS(F, C_K)}$$

Relevancy Score (RS): The other method for creating task-directed LSI representations uses term weights to emphasize the importance of particular terms before applying the SVD. It showed that using an inverse document frequency (IDF) weighting on the term by document matrix before applying the SVD led to 40% average improvement on a set of standard IR test sets. We also found IDF weighting to lead to large improvements in topic spotting performance and use it in all of the experiments reported here. IDF weighting of low frequency terms and diminishes the importance of high frequency terms so that the SVD is forced to distribute its resources more evenly over all terms. It is based on the assumption that, in general, low frequency terms are better discriminators than high frequency terms. In topic spotting, however, we can tune this general assumption.

D. Text Categorization

K – NN Classifier Algorithm: K-Nearest Neighbor is one of the most popular algorithms for text categorization. Many researchers have found that the k-NN algorithm achieves very good performance in their experiments on different data sets. In pattern recognition, the k-nearest neighbor algorithm (k-NN) is a method for classifying objects based on closest training examples in the feature space. k-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. By simply assigning the property value for the object to be the average of the values of its k nearest neighbors. It can be useful to weight the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. (A common weighting scheme is to give each neighbor a weight of 1/d, where d is the distance to the neighbor. This scheme is a generalization of linear interpolation.

ESTIMATING CONTINUOUS VARIABLES

The k-NN algorithm can also be adapted for use in estimating continuous variables. One such implementation uses an inverse distance weighted average of the k-nearest multivariate neighbors. This algorithm functions as follows:

1. Compute Euclidean or Mahalanobis distance from target plot to those that were sampled.
2. Order samples taking for account calculated distances.
3. Choose heuristically optimal k nearest neighbor based on RMSE done by cross validation technique.
4. Calculate an inverse distance weighted average with the k-nearest multivariate neighbors.

Naïve Bayesian classification: The Naïve Bayesian classification system is based on Bayes' rule and works as follows. There are classes, say C_k for the data to be classified into. Each class has a probability $P(C_k)$ that represents the prior probability of classifying an attribute into C_k; the values of $P(C_k)$ can be estimated from the training dataset. For n attribute values, v_j, the goal of classification is clearly to find the conditional probability $P(C_k \mid v_1 \wedge v_2 \wedge \ldots \wedge v_n)$. By Bayes' rule, this probability is equivalent to

$$\frac{P(v_1 \wedge v_2 \wedge \ldots \wedge v_n \mid C_k)P(C_k)}{P(v_1 \wedge v_2 \wedge \ldots \wedge v_n)}$$

For classification, the denominator is irrelevant, since, for given values of the v_j, it is the same regardless of the value of C_k. The central assumption of Naïve Bayesian classification is that, within each class, the values v_j are all independent of each other. Then by the laws of independent probability,

For classification, the denominator is irrelevant, since, for given values of the v_j, it is the same regardless of the value of C_k. The central assumption of Naïve Bayesian classification is that, within each class, the values v_j are all independent of each other. Then by the laws of independent probability,

$P(v_i \mid \{\text{all the other values of } v_j\}, C_k) = P(v_i \mid C_k)$ and therefore

$P(v_1 \wedge v_2 \wedge \dots \wedge v_n \mid Ck) = P(v_1 \mid C_k)P(v_2 \mid C_k)\dots P(v_n \mid C_k)$.
Each factor on the right-hand side of this equation can be determined from the training data, because (for an arbitrary v_i),
$P(v_i \mid C_k) \approx [\#(v_i \wedge C_k)] / [\#(C_k)]$
where "#" represents the number of such occurrences in the training set data. Therefore, the classification of the test set can now be estimated by,
$P(C_k \mid v_1 \wedge v_2 \wedge \dots \wedge v_n)$ which is proportional to
$P(C_k)\, P(v_1 \mid C_k)\, P(v_2 \mid C_k)\, P(v_3 \mid C_k) \dots P(v_n \mid C_k)$.

As mentioned above, the central assumption in Naïve Bayesian classification is that given a particular class membership, the probabilities of particular attributes having particular values are independent of each other.

IV. EXPERIMENTAL RESULTS

Even though these numbers are not comparable to other results since a subset and not the complete Reuters 21578 ModApt´e split was used, they provide still interesting Insights. Especially the fact, that for the same weighting function and the same dimensionality, it happens that, e.g., the breakeven value is higher compared to another function but the eleven-point precision is lower, compared to the same function.

It also shows that" MSF" could be an interesting alternative to chi-square and information gain, not only for feature selection in text classification, but also to weight the importance of features in other classification tasks.

Precision-recall breakeven point

The properties or the expected behaviours of text categorization/information retrieval systems can vary. For example for one system it is better to return mostly correct answers, while in another it is better to cover more true positives. There is a trade off between precision and recall: if a classifier says "True" to every category for every document, then it receives perfect recall, but very low precision. However it can be easily seen that if a classifier says "False" for every category, except one which is correct (TP = 1,FP = 0) then it will have a precision equal to 1 but a very low recall.

That is why it makes comparison between systems easier if the system is characterized by a single value, the breakeven point (BEP), which is the point at which precision equals recall. This can be achieved by tuning the parameters of the system. When there is no such point (because TP, FP and FN are natural numbers) the average of the nearest precision and recall is used, and is called interpolated BEP. For example in ranking categorization models for each class an optimal τ_i CSV threshold has to be determined such that $P \simeq R$. If $CSV_i(d_j) \geq \tau_i$ then the classifier says "True", otherwise says "False".

11-point average precision

The 11-point average precision is another measure for representing performance with a single value. For every category the τ_i CSV threshold is repeatedly tuned such that allow the recall to take the values $0.0, 0.1, \dots, 0.9, 1.0$. At every point the precision is calculated and at the end the average over these eleven values is returned [Sebastiani02]. The retrieval system must support ranking policy.
The following detailed algorithm for the calculation of this value. The precision and recall values for a given document and a threshold is calculated as

$$P = \frac{\text{categories found and correct}}{\text{total categories found}}; \quad R = \frac{\text{categories found and correct}}{\text{total categories correct}}$$

1. For each document calculate the precision and recall at each position in the ranked list where a correct category is found.
2. For each interval between thresholds $0.0, 1.0, \dots, 0.9, 1.0$ use the highest precision value in that interval as the "representative" precision value at the left boundary of this interval.
3. For the recall threshold of 1.0 the "representative" precision is either the exact precision value if such point exists, or the precision value at the closest point in terms of recall. If the interval is empty we use the default precision value of 0.
4. Interpolation: At each of the above recall thresholds replace the "representative" precision using the highest score among the "representative" precision values at this threshold and the higher thresholds.
5. Per-interval averaging: Average per-document data points over all the test documents at each of the above recall thresholds respectively. This step results in 11 per-interval precision scores.
6. Global averaging: Average of the per-interval average precision scores to obtain a single-numbered performance average. The resulting value is called the 11-point average precision.

Table 1 Comparison of P/R using existing with proposed system

Algorithms	Recall									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Existing	0.61	0.45	0.41	0.38	0.35	0.31	0.28	0.26	0.25	0.24
KNN	0.85	0.72	0.64	0.57	0.46	0.40	0.38	0.31	0.29	0.25
NB	0.91	0.85	0.72	0.64	0.55	0.46	0.41	0.39	0.33	0.31

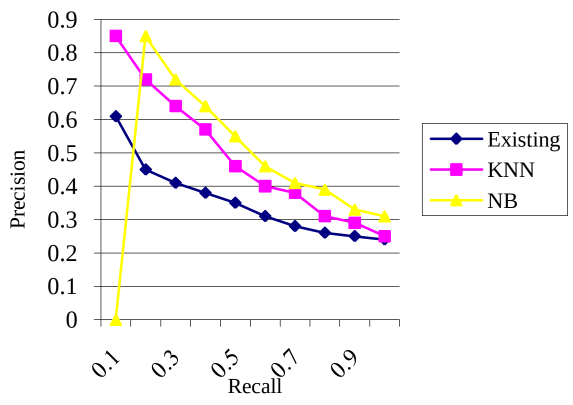

Fig. 1 Compare precision and recall

V. CONCLUSION

In the selection process, each feature (term or single word) is assigned with a score according to a score-computing function. Then those with higher scores are selected. These mathematical definitions of the score-computing functions are often defined by some probabilities which are estimated by some statistic information in the documents across different categories. Text classification is a supervised technique that uses labelled training data to learn the classification system and then automatically classifies the remaining text using the learned system. Classification plays a vital role in many information management and retrieval tasks. Classification includes different parts such as text processing, feature extraction, feature vector construction and final classification. Here apply machine learning methods for classification. In this regard, we first try to exert some text pre-process in different dataset, and then we extract a feature vector for each new document by using feature weighting and feature selection algorithms for enhancing the text classification accuracy. After that we train our classifier by Naïve Bayesian (NB) and support vector machine (KNN) algorithms. In Experiments, although both algorithms show acceptable results for text classification.

REFERENCES

[1] W. Lam, M. Ruiz, and P. Srinivasan, “Automatic text categorization and its application to text retrieval,” IEEE Trans. Knowl. Data Eng., vol. 11, no. 6, pp. 865–879, Nov./Dec. 1999.
[2] F. Sebastiani, “Machine learning in automated text categorization,” ACM Comput. Surveys, vol. 34, no. 1, pp. 1–47, 2002.
[3] G. Forman, “An extensive empirical study of feature selection metrics for text classification,” The J. Mach. Learn. Res., vol. 3, pp. 1289–1305, 2003.
[4] H. Liu and L. Yu, “Toward integrating feature selection algorithms for classification and clustering,” IEEE Trans. Knowl. Data Eng., vol. 17, no. 4, pp. 491–502, Apr. 2005.
[5] P. M. Baggenstoss, “Class-specific feature sets in classification,” IEEE Trans. Signal Process., vol. 47, no. 12, pp. 3428–3432, Dec. 1999. [6] P. M. Baggenstoss, “The pdf projection theorem and the class-specific method,” IEEE Trans. Signal Process., vol. 51, no. 3, pp. 672–685, Mar. 2003.
[6] A. McCallum and K. Nigam, “A comparison of event models for naïve Bayes text classification,” in Proc. Workshop Learn. for Text Categorization, 1998, vol. 752, pp. 41–48.
[7] V. Kecman, Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models. Cambridge, MA, USA: MIT Press, 2001.
[8] D. Koller and M. Sahami, “Hierarchically classifying documents using very few words,” in Proc. 14th Int. Conf. Mach. Learn., 1997, pp. 170–178.
[9] Y. H. Li and A. K. Jain, “Classification of text documents,” The Comput. J., vol. 41, no. 8, pp. 537–546, 1998.
[10] T. Joachims, “Text categorization with support vector machines: Learning with many relevant features,” in Proc. 10th Eur. Conf. Mach. Learn., 1998, pp. 137–142.
[11] B. Tang and H. He, “ENN: Extended nearest neighbor method for pattern recognition [research frontier],” IEEE Comput. Intell. Mag., vol. 10, no. 3, pp. 52–60, Aug. 2015.
[12] S. Eyheramendy, D. D. Lewis, and D. Madigan, “On the naive Bayes model for text categorization,” in Proc. 9th Int. Workshop Artif. Intell. Statist., 2003, pp. 332–339.
[13] L. Galavotti, F. Sebastiani, and M. Simi, “Experiments on the use of feature selection and negative evidence in automated text categorization,” in Proc. 4th Eur. Conf. Res. Adv. Technol. Digit. Libraries, 2000, pp. 59–68.
[14] B. Tang, S. Kay, and H. He, “Toward optimal feature selection in naïve Bayes for text categorization,” Preprint, arXiv:1602.02850 [stat.ML], 2016.
[15] X. Fu and L. Wang, “A GA-based RBF classifier with class-dependent features,” in Proc. Congress Evol. Comput., 2002, vol. 2, pp. 1890–1894.
[16] L. Wang, N. Zhou, and F. Chu, “A general wrapper approach to selection of class-dependent features,” IEEE Trans. Neural Netw., vol. 19, no. 7, pp. 1267–1278, Jul. 2008.
[17] S. Kay, “Asymptotically optimal approximation of multidimensional pdf’s by lower dimensional pdf’s,” IEEE Trans. Signal Process., vol. 55, no. 2, pp. 725–729, Feb. 2007.
[18] B. Tang, H. He, Q. Ding, and S. Kay, “A parametric classification rule based on the exponentially embedded family,” IEEE Trans. Neural Netw. Learn. Syst., vol. 26, no. 2, pp. 367–377, Feb. 2015.
[19] D. Cai, X. He, and J. Han, “Document clustering using locality preserving indexing,” IEEE Trans. Knowl. Data Eng., vol. 17, no. 12, pp. 1624–1637, Dec. 2005.
[20] D. Cai, Q. Mei, J. Han, and C. Zhai, “Modeling hidden topics on document manifold,” in Proc. 17th ACM Conf. Inf. Knowl. Manage., 2008, pp. 911–920.

BIOGRAPHIES

B. Senthil Kumar graduated from the Department of Computer Science, Bharathiar University, CBE, in 2004. He obtained Post Graduate in 2007 and M.Phil. in CS from Bharathiar University, CBE in 2009. He is now a Doctoral Student (Part Time) in the Department of Computer Science, Bharathiar University. His current field of research is Data Mining and Health Informatics. He has published papers in international journals. He is also an Assistant Professor in the Department of Computer Science, Sree Narayana Guru College, CBE.

Bhavitha Varma .E graduated from the Department of Information Technology, Bharathiar University, AJK college of Art and Science, in 2011.She was completed her Post Graduation in 2014 in CS from Calicut University, VTB College. Now pursuing M Phil. in CS from Bharathiar University