Different Type of Feature Selection for Text Classification

M.Ramya[#1], J.Alwin Pinakas[#2]
(Institute Of Information Management,Kgisl / Bharathiar University,India)

ABSTRACT *– Text categorization is the task of deciding whether a document belongs to a set of pre specified classes of documents. Automatic classification schemes can greatly facilitate the process of categorization. Categorization of documents is challenging, as the number of discriminating words can be very large. Many existing algorithms simply would not work with these many numbers of features. For most text categorization tasks, there are many irrelevant and many relevant features. The main objective is to propose a text classification based on the features selection and pre-processing thereby reducing the dimensionality of the Feature vector and increase the classification accuracy. In the proposed method, machine learning methods for text classification is used to apply some text preprocessing methods in different dataset, and then to extract feature vectors for each new document by using various feature weighting methods for enhancing the text classification accuracy. Further training the classifier by Naive Bayesian (NB) and K-nearest neighbor (KNN) algorithms, the predication can be made according to the category distribution among this k nearest neighbors. Experimental results show that the methods are favorable in terms of their effectiveness and efficiency when compared with other.*

Keywords– *Feature selection, K-Nearest Neighbor, Naïve Bayesian, Text classification.*

I. INTRODUCTION

Automated text classification is a particularly challenging task in modern data analysis, both from an empirical and from a theoretical perspective. This problem is of central interest in many internet applications, and consequently it has received attention from researchers in such diverse areas as information retrieval, machine learning, and the theory of algorithms. Challenges associated with automated text categorization come from many fronts: one must choose an appropriate data structure to represent the documents; one must choose an appropriate objective function to optimize in order to avoid over fitting and obtain good generalization and one must deal with algorithmic issues arising as a result of the high formal dimensionality of the data.

Feature selection, i.e., selecting a subset of the features available for describing the data before applying a learning algorithm, is a common technique for addressing this last challenge. It has been widely observed that feature selection can be a powerful tool for simplifying or speeding up computations, and when employed appropriately it can lead to little loss in classification quality. Nevertheless, general theoretical performance guarantees are modest and it is often difficult to claim more than a vague intuitive understanding of why a particular feature selection algorithm performs well when it does. Indeed, selecting an optimal set of features is in general difficult, both theoretically and empirically; hardness results are known, and in practice greedy heuristics are often employed.

.

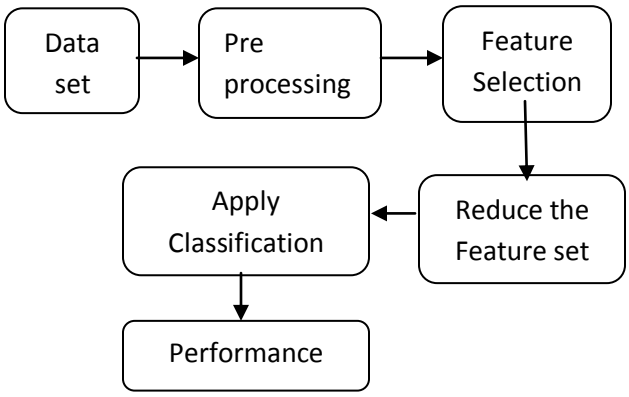

Fig 1: Text Classification Framework

1. Related Work

Several researchers have emphasized on the issue of redundant attributes, as well as advantages of feature selection for the Naïve Bayesian Classifier, not only for induction learning. Pazzani explores the methods of joining two (or more) related attributes into a new compound attribute where the attribute dependencies are present. Another method, Boosting

on Naïve Bayesian classifier has been experimented by applying series of classifiers to the problem and paying more attention to the examples misclassified by its predecessor. However, it was shown that it fails on average in a set of natural domain. Langley and Sage use a wrapper approach for the subset selection to only select relevant features for NB. Cardie uses the attributes from decision trees in combination with nearest neighbor methods. And in a domain for discovering patterns in EEG-signals, Kubat, Flotzinger, and Pfurtscheller tried the use of Decision tree in feature selection for Naïve Bayesian classifier. And recently, Augmented Bayesian Classifiers was introduced as another approach where Naïve Bayes is augmented by the addition of correlation arcs between attributes.

II. PROPOSED METHODOLOGY

1. Preprocessing

The goal behind preprocessing is to represent each document as a feature vector, that is, to separate the text into individual words. In the proposed classifiers, the text documents are modeled as transactions. Choosing the keyword that is the feature selection process, is the main preprocessing step necessary for the indexing of documents.

This step is crucial in determining the quality of the next stage, that is, the classification stage. It is important to select the significant keywords that carry the meaning, and discard the words that do not contribute to distinguishing between the documents.

2. Stop Word Removal

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.

3. Stemming Algorithm

Stemming is the process of grouping words that share the same morphological root. E.g. ”game” and ”games” are stemmed to the root ”game”. The suitability of stemming to Text Classification is controversial. In some examinations, Stemming has been reported to hurt accuracy. However, the recent tendency is to apply it, since it reduces both the dimensionality of the term space and the stochastic dependence between terms.

3.1 Porter Stemming

Porter stemming algorithm is a process for removing the commoner morphological ending words in English [8]. Rules in porter stemming algorithm are separated into five distinct steps:

1) Gets rid of plurals and -ed or -ing. eg-> caress ponies -> ponities -> ti caress -> caress cats -> cat

2) Turns terminal y to i when there is another vowel in the stem. eg happy->happi

3) Maps double suffices to single ones. so -ization (= -ize plus -ation) maps to -ize etc.

4) Deals with -ic-, -full, -ness etc. similar strategy to step3.

5) Takes off -ant, -ence etc.

3.2 Lancaster Stemming

The new debugging options helped to solve the mystery of why the original rules generated the stem "abud" from "abusively":

<abusively> 100->abusive 13->abusiv 94->abuj 27->abud

Affix removal conflation techniques are referred to as stemming algorithms and can be implemented in a variety of different methods. All remove suffices and/or prefixes in an attempt to reduce a word to its stem.. The algorithms that are discussed in the following sections, and those that will be implemented in this project, are all suffix removal stemmers.

III. FEATURE WEIGHTING

For many machine learning algorithms it is necessary
to reduce the dimensionality of the feature space, if the original dimensionality of the space is very high. In most of the cases this improves not only the

performance but also the accuracy of the classification itself. Term Weighting is the process of assigning values to all the terms in the corpus according to their importance for the actual classification part. Here, importance is defined as the ability of the term to distinguish between different categories in the corpus. Usually, the more important a term is the higher is the assigned weight value.

1. Document Frequency (DF)

Simply measures in how many documents the word appears. Since it can be computed without class labels, it may be computed over the entire test set as well. Selecting frequent words will improve the chances that the features will be present in future test
cases. It is defined as

$$DF = \sum_{i=1}^{m} (A_i)$$

2. Mutual Information (MI)

The mutual information of two random variables is a quantity that measures the mutual dependence of the two random variables. MI measures how much information the presence/absence of a term contributes to making the correct classification decision.

$$MI(F, C_k) = \sum_{v_f \in \{1,0\}} \sum_{vck \in \{1,0\}} p(F = v_f, ck = v_{ck}) \ln \frac{P(F = v_f, Ck = V_{ck})}{P(F = V_f) P(Ck = V_{ck})}$$

3. Information Gain (IG)

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.

It is defined by following expression

$$IG(t) = -\sum_i \Pr(C_i) \log \Pr(C_i) + \Pr(t) \sum_i \Pr(C_i|t) \log \Pr(C_i|t) + \Pr(t) \sum_i \Pr(C_i|t) \log Pr(C_i|t)$$

It is frequently used as a term goodness criterion in machine learning. It measures the number of bits required for category prediction by knowing the presence or the absence of a term in the document.

4. X2 □Statistic (CHI)

Feature Selection by Chi - square testing is based on Pearson's X 2 (chi square) tests. The Chi square test of independence helps to find out the variables X and Y are related to or independent of each other. In feature selection, the Chi - square test measures the independence of a feature and a category. The null-hypothesis here is that the feature and category are completely independent. It is defined by,

$$x^2(F, C_k) = \frac{N \times ((N_{F, Ck} \times N_{\bar{F}, \bar{Ck}}) - (N_{F, \bar{Ck}} \times N_{\bar{F}, Ck}))^2}{N_F \times N_{\bar{F}, Ck} \times N_{\bar{Ck}}}$$

5. Ngl Coefficient

The NGL coefficient is a variant of the Chi square metric. It was originally named a `correlation coefficient', 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.

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

6. Term Frequency Document Frequency

The tf–idf weight is a method based on the term frequency combined with the document frequency threshold, it is defined as,

$$TFDF(F) = (n_1 \times n_2 + c(n_1 \times n_3 + n_2 \times n_3))$$

7. GSS Coefficient

The GSS coefficient was originally presented in [GSS00] as a `simplified chi square function'. We

follow [Seb02] and name it GSS after the names on the inventors Galavotti, Sebastiani, and Simi.

$$\mathbf{Gss}(\mathbf{F},\mathbf{c_k}) = \mathbf{N}_{F,Ck}\mathbf{N}_{\overline{F},\overline{Ck}} - N_{F,\overline{Ck}}N_{\overline{F},Ck}$$

IV. TEXT CATEGORIZATION

With the rapid growth of online information, there is a growing need for tools that help in finding, filtering and managing the high dimensional data. Automated text categorization is a supervised learning task, defined as assigning category labels to new documents based on likelihood suggested by a training set of labeled documents. Real-world applications of text categorization often require a system to deal with tens of thousands of categories defined over a large Taxonomy. Since building these text classifiers by hand is time consuming and costly, automated text categorization has gained importance over the years.

1. K–Nearest Neighbor Classifier Algorithm

K-Nearest Neighbor is one of the most popular algorithms for text categorization. *K-nearest* neighbor algorithm (*k*-NN) is a method for classifying objects based on closest training examples in the space. The working of KNN can be detailed as follows first the test document has to be classified the KNN algorithm searches the nearest neighbors among the training documents that are pre classified. The ranks for the K nearest neighbors based on the similarity scores are calculate using some similarity measure such as Euclidean distance measure etc., The distance between two neighbors using Euclidean distance can be found using the given formula the categories of the test document can be predicted using the ranked scores. The classification for the input pattern is the class with the highest confidence; the performance of each learning model is tracked using the validation technique called cross validation. The cross validation technique is used to validate the pre determined metric like performance and accuracy.

$$Dist(X,Y) = \sqrt{\sum_{i=1}^{D}(Xi - Yi)^2}$$

While using kNN algorithm, after k nearest neighbors is found, several strategies could be taken to predict the category of a test document based on them. A fixed k value is usually used for all classes in these methods, regardless of their different distributions. Equation (1) and (2) below are two of the used strategies of this kind method.

$$y(d_i) = \arg max_k \sum_{x_j \in KNN} y(x_j, c_k)$$

$$y(d_i) = arg max_k \sum_{x_j \in KNN} Sim(d_i, x_j) y(x_j, c_k)$$

where d_i is a test document, x_i is one of the neighbors in the training set,$y(x_j,c_k) \in \{0,1\}$ indicates whether x_j belongs to class c_k ,$sim(d_i,x_j)$ and is the similarity function for d_i and x_j. Equation (1) means that the predication will be the class that has the largest number of members in the k nearest neighbors; whereas equation (2) means the class with maximal sum of similarity will be the winner.

2. Naive Bayesian Classifier Algorithm

The Naive Bayes classifiers are known as a simple Bayesian classification algorithm. It has been proven very effective for text categorization. Regarding the text categorization problem, a document d 2 D corresponds to a data instance, where D denotes the training document set. The document d can be represented as a bag of words. Each word w 2 d comes from a set W of all feature words. Each document d is associated with a class label c 2 C, where C denotes the class label set. The Naive Bayes classifiers estimate the conditional probability P(c|d) which represents the probability that a document d belongs to a class c. Using the Bayes rule, we have

$$P(c|d) \propto P(c) \,.P(d|c)$$

The key assumption of Naive Bayes classifiers is that the words in the documents are conditionally independent given the class value, so that

$$P(c|d) \propto P(c)\Pi_{w \in d} P(w|c)$$

A popular way to estimate P(w|c) is through Laplacian smoothing:

$$P(w|c) = \frac{1 + n(w.c)}{|w| + n(c)}$$

where n(w, c) is the number of the word positions that are occupied by w in all training examples whose class value is c. n(c) is the number of word positions whose class value is c. Finally, |W| is the total number of distinct words in the training set.

V. PERFORMANCE METRIC

The evaluation of a classifier is done using the precision and recall measures .To derive a robust measure of the effectiveness of the classifier It is able to calculate the breakeven point, the 11-point precision and ”average precision” . to evaluate the classification for a threshold ranging from 0 (recall = 1) up to a value where the precision value equals 1 and the recall value equals 0, incrementing the threshold with a given threshold step size. The breakeven point is the point where recall meets precision and the eleven point precision is the averaged value for the precision at the points where recall equals the eleven values 0.0, 0.1, 0.2... 0.9, 1.0. ”Average precision” refines the eleven point precision, as it approximates the area”below” the precision/recall curve.

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 $0.0, 0.1, \ldots, 0.9, 1.0$. 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.

VI. EXPERIMENTAL RESULTS

Data Set 1: Self Made

For the development used a small self-made corpus that contains standard categories such as ”Science”, ”Business”, ”Sports”, ”Health”, ”Education”, ”Travel”, and ”Movies”. It contains around 150 documents with the above mentioned categories.

Data Set 2: The Reuters 21578 corpus

The second corpus included for the development is Reuters 21578 corpus. The corpus is freely available on the internet (Lewis 1997). Uses an XML parser, it was necessary to convert the 22 SGML documents to XML, using the freely available tool SX (Clark 2001). After the conversion I deleted some single characters which were rejected by the validating XML parser as they had decimal values below 30. This does not affect the results since the characters would have been considered as whitespaces anyway.

Table 1: Performances of two classification algorithms

Data Set	Knn				SVM			
	Micro-Avg.	Macro-Avg.			Micro-Avg.	Macro-Avg.		
	Pre=Rec= F1	Pre	Rec	F1	Pre=Rec= F1	Pre	Rec	F1
Reuters	71.69	67.10	66.57	66.02	72.35	70.74	77.24	77.96
20 news group	70.12	67.88	67.56	66.29	75.24	75.17	78.49	78.11
Avg.	70.90	67.49	66.43	66.15	73.79	72.95	77.86	78.03

To evaluate the effectiveness of category assignments by classifiers to documents, the standard precision, recall, and F 1 measure are used here. Precision is defined to be the ratio of correct assignments by the system divided by the total number of the system’s assignments. Recall is the ratio of correct assignments by the system divided by the total number of correct assignments.

These scores can be computed for the binary decisions on each individual category first and then be averaged over categories. Or, they can be computed globally over all the n*m binary decisions where n is the number of total test documents, and m is the number of categories in consideration. The former way is called macro-averaging and the latter

micro-averaging. It is understood that the micro-averaged scores (recall, precision, and F 1) tend to be dominated by the classifier's performance on common categories, and that the macro-averaged scores are more influenced by the performance on rare categories.

VII. CONCLUSION

Analyzed the text classification using the Naive Bayesian and K-Nearest Neighbor classification. The methods are favorable in terms of their effectiveness and efficiency when compared with other classifier such as SVM. The advantage of the proposed approach is classification algorithm learns importance of attributes and utilizes them in the similarity measure. In future the classification model can be build that analyzes terms on the sentence, document.

REFERENCES

[1] A Fuzzy Self-Constructing Feature Clustering Algorithm for Text Classification" Jung-Yi Jiang, Ren-Jia Liou, and Shie-Jue Lee, Member, IEEE TRANS ON Knowledge and Data Eng.,Vol 23,No.3,March 2011

[2] J. Yan, B. Zhang, N. Liu, S. Yan, Q. Cheng, W. Fan, Q. Yang, W. Xi,and Z. Chen, "Effective and Efficient Dimensionality Reduction for Large-Scale and Streaming Data Preprocessing," IEEE Trans.Knowledge and Data Eng., vol. 18, no. 3, pp. 320-333, Mar. 2006.

[3] H. Li, T. Jiang, and K. Zang, "Efficient and Robust Feature Extraction by Maximum Margin Criterion," T. Sebastian, S.Lawrence, and S. Bernhard eds. Advances in Neural Information Processing System, pp. 97-104, Springer, 2004.

[4] D.D. Lewis, "Feature Selection and Feature Extraction for Text Categorization," Proc. Workshop Speech and Natural Language,pp. 212-217, 1992.

[5]http://kdd.ics.uci.edu/databases/ reuters21578/reuters21578.html. 2010.

[6] H. Kim, P. Howland, and H. Park, "Dimension Reduction in Text Classification with Support Vector Machines," J. Machine Learning Research, vol. 6, pp. 37-53, 2005.

[7] F. Sebastiani, "Machine Learning in Automated Text Categorization," ACM Computing Surveys, vol. 34, no. 1, pp. 1-47, 2002.

[8] H. Park, M. Jeon, and J. Rosen, "Lower Dimensional Representation of Text Data Based on Centroids and Least Squares," BIT Numerical Math, vol. 43, pp. 427-448, 2003.