AN EFFICIENT TEXT CLASSIFICATION USING KNN AND NAIVE BAYESIAN

J.Sreemathy
Research Scholar
Karpagam University
Coimbatore, India

P. S. Balamurugan
Research Scholar
ANNA UNIVERSITY
Coimbatore, India

Abstract – **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. Text classification is the process of assigning a document to one or more target categories, based on its contents. 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 classifier such as SVM.**

Keywords– Text classification; Feature selection; K-Nearest Neighbor; Naïve Bayesian

I. INTRODUCTION

The dimensionality of a feature vector is the major problem in Text classification. The dimensionality of features is reduced by the technique called preprocessing. The preprocessing helps to identify the informative features in order to improve the accuracy of the classifier. Unstructured database is preprocessed by stopping keyword removal and stemming removes the obstacle words in order to improve the accuracy of the classifier. There are some noise reduction techniques that work only for k-NN that can be effective in improving the accuracy of the classifier. Naive Bayes can be used for both binary and multiclass classification problems.

II. FEATURE SELECTION

Feature selection is important task in classifying the text which improves the classification accuracy by eliminating the noise features from various corpuses. Feature selection is defined as the process of selecting a subset of relevant features for building robust learning models. Feature selection method has two main reasons. Initially, it makes the training and applying a classifier to more efficient by decreasing the dimensionality of the feature vector size.

III. FEATURE WEIGHTING

The feature weighting is the process of assigning a score for each feature i.e., term or a single word using a score computing function. The features that have high scores are selected. The scores are assigned for each feature in a selected subset of relevant features. 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. There are seven different weighting functions implemented.

A. Document Frequency (DF)

DF is the number of documents in which a term occurs. It is defined as

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

B. 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.

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

C. 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.

$$IG(t) = H(C) - H(C|T) = \Sigma\tau,c\ P(C=c,T=\tau)$$

$$\ln [P(C=c, T=\tau)/P(C=c)P(T=\tau)].$$

Here, τ ranges over {present, absent} and c ranges over {c+, c–}.

D. χ^2 Statistic (CHI)

Feature Selection by Chi - square testing is based on Pearson's χ^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,

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

E. NGL coefficient

The NGL coefficient presented in [NGL97] is a variant of the Chi square metric. It was originally named a `correlation coefficient', but we follow Sebastiani [Seb02] and 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.

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

F. 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,

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

G. 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.

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

IV. TEXT CLASSIFICATION PROCESS

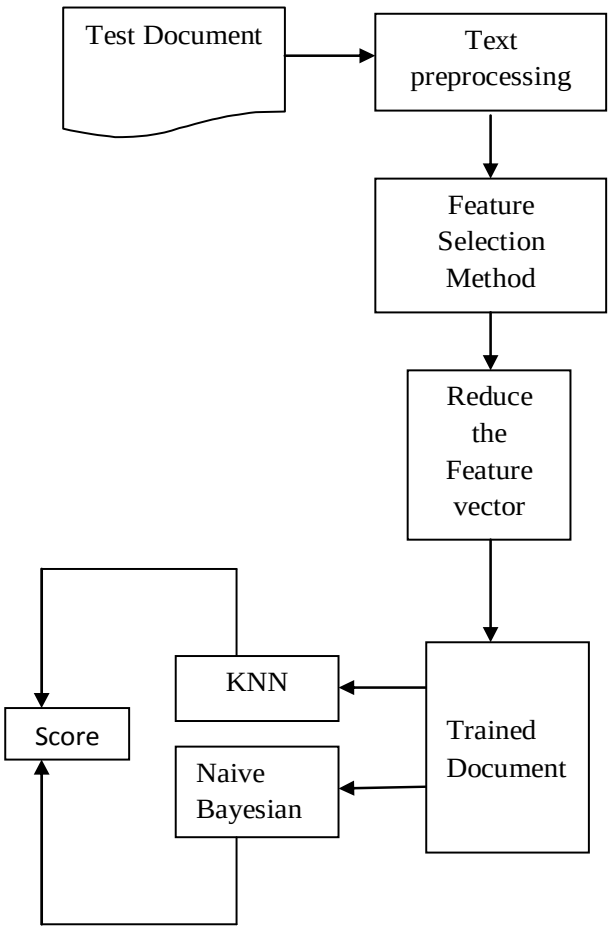

Figure 1. Block diagram for text classification

V. CLASSIFIERS AND DOCUMENT COLLECTIONS

We selected two classifiers for the Text Classification

- K-Nearest Neighbors
- Naive Bayesian

A. K – Nearest Neighbor Classifier

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

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

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.

B. Naive Bayesian Classifier

The naive Bayesian classifier is the probability based classifier, based on the features independent probability value is calculated for each and every model. The naïve Bayesian classifier construct a set of class i.e., set of category label and probability for that category. The scoring is done by ranking each category probabilities for every test document rather than a training document for every test document. The classification for the instance is the class with the highest probability value.

VI. 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.

Table 1. Precision Recall measure across different classifiers

Corpus	**Self made**		**Reuters 21578**	
Terms	***Recall***	***Precision***	***Recall***	***Precision***
SVM	58.8%	75.4%	88.6%	57.9%
KNN	88.8%	80.4%	95.9%	85.5%
Naive Bayesian	90.2%	93.1%	94.6%	95.5%

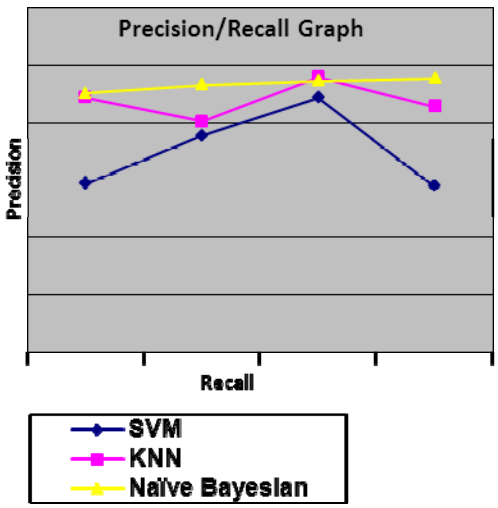

Figure 2. Performance chart of three different classifiers

VII. EXPERIMENTAL RESULTS

A. 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.

B. 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.

VIII. 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, the classification algorithm learns importance of attributes and utilizes them in the similarity measure. In future the classification model can be build, which analyzes terms on the concept sentence in 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] 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
[6] .html. 2010.
[7] 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.
[8] F. Sebastiani, “Machine Learning in Automated Text Categorization,” ACM Computing Surveys, vol. 34, no. 1, pp. 1-47, 2002.
[9] 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.

AUTHORS PROFILE

Sreemathy J received BE degree at 2007 in Computer Science & Engineering from Park College of Engineering and Technology, Coimbatore and currently pursuing M.E. Computer Science & Engineering at Karpagam University, Coimbatore.

Prof. P.S.Balamurugan received BE degree in 2003 from Bharathiyar University Coimbatore and ME in 2005 from Anna University. Currently he is pursuing his Doctoral degree at Anna University, Coimbatore.