ISSN:0975-9646

Suresh Kumar et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (4) , 2015, 3742-3745

Enhancing Text Classification by Stochastic Optimization method and Support Vector Machine

Suresh Kumar [1], Dr. Shivani Goel[2]

[1,2]*CSE Department, Thapar University, Patiala, INDIA*

***Abstract*— Text Classification, also known as text categorization, is the task of automatically allocating unlabeled documents into predefined categories. Text Classification means allocating a document to one or more categories or classes. The ability to accurately perform a classification task depends on the representations of documents to be classified. Text representations transform the textural documents into a compact format. Text Classification plays an important role in information mining, summarization, text recovery and question-answering. It uses several tools from information retrieval (IR) and Machine Learning. Here we are reviewing the effectiveness of different supervised and unsupervised learning approaches in text classification.**

Keywords- Text Mining, Text Classification, Feature Extraction,Term Weighting, Linear SVC, SGD, K-Means with cosine similarity.

I. INTRODUCTION

With the rapid growth of online information, effective retrieval of some particular information is difficult without good indexing and summarization of document content. Text Classification may be the solution to effectively handle and organize such huge text collections. Text Classification is the process of automatically grouping of documents into some predefined categories. The ability to accurately perform a classification task depends on the representation of documents to be classified. In text categorization, text representations transform the content of textual document into a compact format so that documents can be recognized and classified by a classifier. A classifier is a system that repeatedly classifies texts into one of the discrete set of predefined categories. For example, for email management one could benefit from a system that classifies incoming messages as important or unimportant. One of the main theme sustaining text mining is transforming text into numerical vectors i.e. text representations as shown in figure 1.

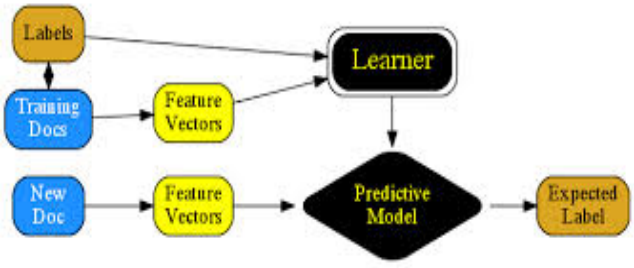

Figure 1: Representation of the flow of learning in Text

Feature selection is a standard procedure for dimensionality reduction. For text classification task an evaluation function is used that is applied in single term for selecting feature subset. After selecting feature subset all terms are sorted and evaluated independently. The best feature subset is determined by predefined threshold. Document frequency (DF) thresholding, information gain (IG), mutual information (MI), chi-square statistic (CHI) are various commonly used feature selection methods in text classification. In information retrieval, documents are generally identified by set of terms or keywords that are collectively used to represent their contents. A document is represented as a vector in the term spaces in Vector Space Model.

$$d = (w1, w2, w3 - - - - - - w|v|)$$

where |v| = size of vocabulary and, lies between [0,1]. The value of w_i represents that how much the term w_i contributes to the semantics of the document. Vector Space Model is one of the mostly used models for text representations (refer fig 2). Generally text representations include two types of works: indexing and term weighting. Indexing is done to allocate indexing terms for documents whereas term weighting is done to assign weight to each term of the document which measures the importance of that term. Presently, there are many term weighting methods which are used for text classifications. Text classification has borrowed the term weighting schemes from IR (information retrieval) field, such as tern frequency (TF), TF-IDF (inverse document frequency) and its variants. Feature representation is a transformation method that allows documents to be interpreted by classifiers and this method is also called as Term Weighting as shown in table 1.

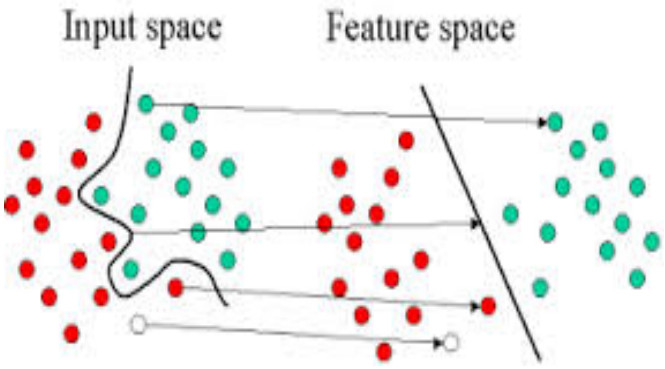

Figure 2: Input and feature space in text

Table 1:Term Weighting Methods

Method	Description
Binary	Boolean Logic Representation 1= Present, 0 = Not Present
TF(Term Frequency)	Frequency of a term in a document i.e no of times the term appears in a document.
DF(Document Frequency)	Frequency of term in collection of documents.

II. LITERATURE SURVEY

There are different indexing methods in text classification(refer fig. 3). **Wen Zhang et.al** studied the comparison of the effectiveness of different indexing methods in text classification like TF_IDF, Latent Semantic Indexing (LSI) and multi-word for text representation[1]. An experimental result demonstrated that in text classification, LSI performed very well than other methods in both document collections. Also, while retrieving English documents LSI showed the best performance. This outcome showed that LSI had both favorable semantic and statistical quality and was different with the claim that LSI cannot produce discriminative power for indexing.

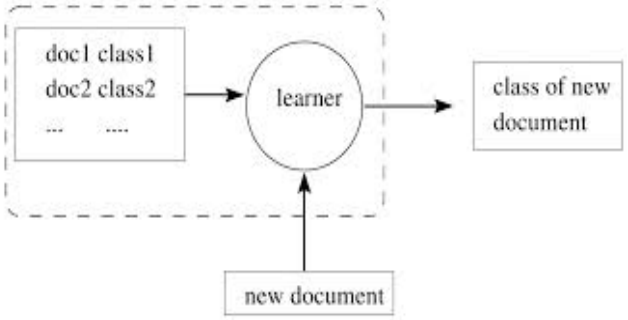

Figure 3: Classification of Documents

Vishwanath Bijalwan et.al first categorized the documents for classification using K nearest neighbor (KNN) based machine learning and then returned the most relevant documents[1]**.** The authors concluded that KNN showed the maximum accuracy as compared to the Naive Bayes and Term-Graph. The disadvantage of KNN classifier found was that its time complexity was high but it gave a enhanced accuracy than others. In this paper the authors rather than implementing the traditional Term-Graph used with AFOPT, used Term-Graph with other methods. This hybrid approach showed better results than the traditional combination. Finally authors made an information retrieval application using Vector Space Model to give the result of the query entered by the client by showing the relevant document.

Text classification is a difficult task due to its high dimensionality of data. Therefore, efficient method for feature selection is required to improve the performance of text classification. A paper by **Tanmay Basu et. al** presented a new feature selection method for text classification using a supervised term selection approach[3]. In this paper term significance (TS) a feature selection technique was compared with CHI,IG & MI. The proposed approach derived a similarity score between a term and a class and then ranked the terms according to their scores over all the classes. The experimental results showed that the proposed TS could produce better classification accuracy even after removing 90% unique terms.

Youngjoong Ko et al. have tried to improve text classification by efficiently applying class information to a term weighting scheme[4]. The authors purposed a new scheme for multi class text classification. Then it was compared to the TF-IDF and previous methods. As a result the proposed scheme utilized class information for term weighting for text classification and performed consistently on the data sets and KNN and support vector machine (SVM) classifiers.

Aixin Sun et al. proposed a simple, scalable and non-parametric approach for short text classification[5]. This approach mimics human classification process for a piece of short text like tweets, status updates, and comments. It selected the representative words from a given short text as query words. After that it searched for a set of labeled text those best matches the query words. The authors used four approaches and were evaluated to select the query words: TF, TF.IDF, TF.CLARITY and TF.IDF.CLARITY. Experimental results showed (refer fig 4) that TF.CLARITY performs effectively when three or more words were used in a query whereas TF.IDF.CLARITY performed well when one word was used in a query. The improvement became very minor when more than five words were used in a query.

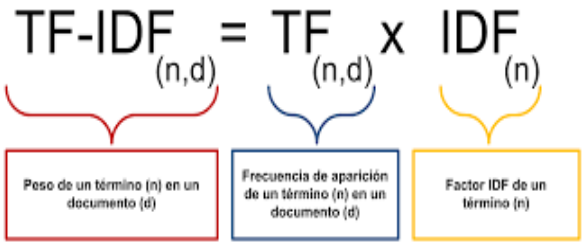

Figure 4: TF-IDF Representation

Chen proposed a new algorithm for short text classification[6]. The author compared the proposed algorithm with the state of the art baseline over web-snippet data set (one open data set) through two type of classifiers: MaxEnt (Maximum Entropy), SVM. The experimental results showed that proposed algorithm performed better and appreciably reduced the classification errors by 16.68% and 20.25% in the same way.

Kiritchenko introduced a learning technique that decreased the effort needed in applying machine learning[7]. Main problems in text classification are lack of labeled data and the cost required for labeling the unlabeled data. In this paper classification was done on E-mail domain with Co-training algorithm that uses unlabeled data along with a small number of labeled examples. In this paper, the author firstly tested SVM classifier on a Labeled edition of unlabeled data and then Naive Bayes classifier was tested. As a result SVM performed very well in comparison with Naive Bayes. Experimental result also showed that the performance of co-training depends on learning method that it used.

Pang et al. proposed a generalized cluster centroid based classifier(GCCC) to use KNN and Rocchio via a clustering algorithm[8]. In this paper, an algorithm was combined with Rocchio and KNN to make a generalized cluster centriod based model respectively to ensure the scalability and applicability of the GCCs model. Experimental results showed that GCCC showed stable and favorable performance than KNN and Rocchio classifier. One drawback of GCCC was that it was more time-consuming than *KNN* and Rocchio.

A relative study for the categorization of verbal autopsy text in three ways i.e. feature representation, effect of reducing features and Machine learning algorithms was done by Danso et al.[9]. The authors exhibit that normalized TF and standard TF-IDF achieved comparable performance across different classifiers. Finally author demonstrated the effectiveness of applying semi-supervised feature reduction approach to increase accuracy and SVM (Support Vector Machine) algorithm found to be the best algorithm than other algorithms.

Larochelle et al. used an individual non-linear Classifier (RBM) for classification[10]. Firstly the classifier RBM (Restricted Boltzmann Machine) was trained through different strategies and then tested with two classifiers i.e. LOG and NNet. In this paper RBM was compared with two different classifiers on multitask datasets. As a result RBM classifier gave best performance on all datasets than other classifiers.

III. PROPOSED WORK

In proposed algorithm unstructured text is firstly preprocessed and TF-IDF is used to give weight to the text. After preprocessing, features are extracted from the text. Then the text is converted into a trainable classifier model using SVC classifier. After training, using SVC a model is generated and testing will be done on it. In testing module also, tokenization and features are extracted as before will be extracted and is tested on trained SVC model that whether it predicts class as trained or not.

The proposed methodology for e-mail spam detection using NLP involves following steps:

Step 1: Tokenization & Stemming:

Tokenization is the process of breaking up the given text into units called tokens. The tokens may be words or numbers or punctuation marks. Tokenization does this task by locating word boundaries. Ending point of a word and beginning of the next word is called word boundaries. Tokenization is also known as word segmentation.

Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes.

Step 2: Vector Model of Text

Vector space model or **term vector model** is an algebraic model for representing text documents (and any objects, in general) as vectors of identifiers, such as, for example, index terms. It is used in information filtering, information retrieval, indexing and relevancy rankings[11].

Step 3: Feature Selection

Feature selection is the process of selecting a subset of the terms occurring in the training set and using only this subset as features in text classification. Feature selection serves two main purposes. First, it makes training set smaller and applying a classifier more efficient by decreasing the size of the effective vocabulary. This is of particular importance for classifiers that, unlike NB, are expensive to train. Second, feature selection often increases classification accuracy by eliminating noise features.

Step 4: Proposed Methodology

Our system consists of three modules:

a) Feature extraction b) Training c) Testing

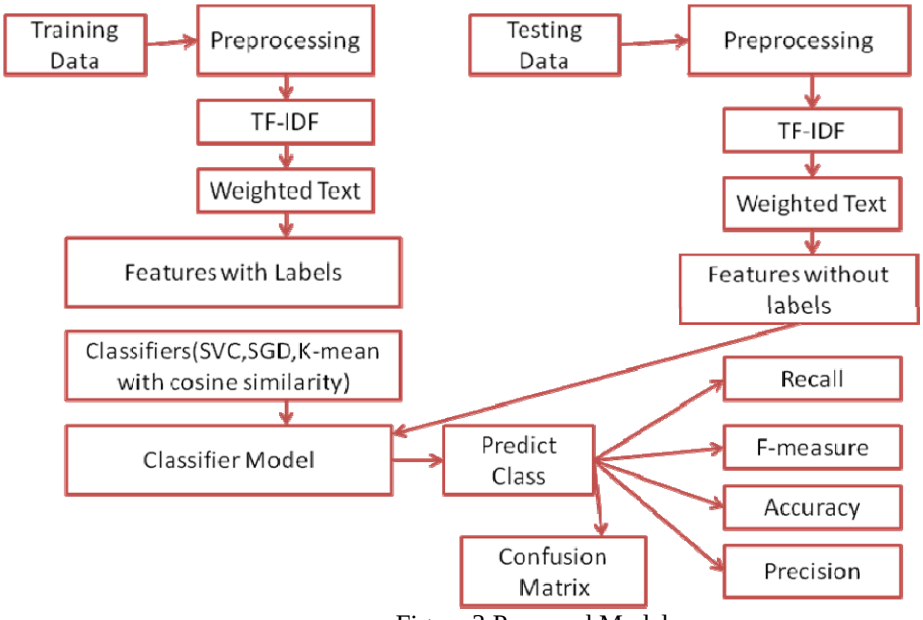

Figure 3:Proposed Model

Table 2: Result for various algorithms

Classifier	KNN	Linear SVC	SGD	K-mean	Multinomial NB	Bernoulli NB	SVM
F-measure	85.61	82.84	82.64	72.23	85.77	87.14	86.78
Accuracy	78.27	86.24	86.27	87.6	84.23	83.4	84.23
Precision	89.23	91.36	91.24	83.6	94.23	88.17	89.92
Recall	89.29	95.64	95.64	96.01	92.34	93.7	84.23

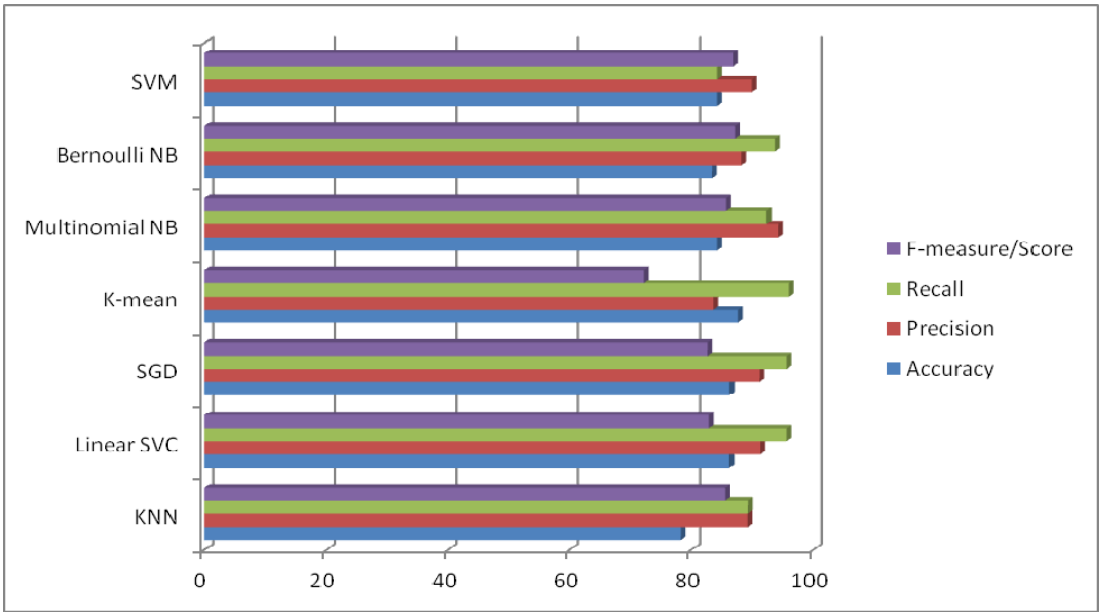

Figure 4: Results for All Algorithms Applied

IV. RESULTS

Table 2 shows various results obtained after applying all steps for classification as discussed above in python language in NLTK(National language toolkit) with 20 newsgroup dataset with four classes.

Above table shows the best result for best two algorithms: Naïve Bayes and SVM. Considering the low false positive ratio, Naïve Bayes performs well as it is easier to implement and has low running time but has less accuracy than Linear SVC and SGD. Hence we conclude that optimization methods perform well and show better results than other classifiers. The values of F-measure, recall, precision and accuracy of predication are shown in figure 4.

V. CONCLUSION AND FUTURE SCOPE

From the experimental results and discussion, it can be concluded that optimization methods like Naïve Bayes and SVM perform well and show better results than other classifiers like SGD, KNN etc. The values of F-measure, recall, precision and accuracy of predication as shown in figure 4 is highest for these two algorithms. In future, a larger dataset can be used for checking the performance of these.

REFERENCES

[1] Vishwanath Bijalwan, Vinay Kumar, Pinki Kumari, Jordan Pascual, "KNN based Machine Learning Approach for Text and Document Mining", *International Journal of Database Theory and Applications*, Vol.7, No.1, 2014, pp. 61-70.

[2] Wen Zhang, Taketoshi Yoshida, Xijin Tang, "A Comparative Study of TF*IDF ,LSI and multi words for text classification", *Expert Systems with Application Journal Elsevier*, Vol. 38, No. 3, 2011, pp. 2758-2765.

[3] Tanmay Basu, C. A. Murthy, "Effective Text Classification by a Supervised Feature Selection Approach", *Proceedings of the 2012 IEEE 12th International Conference on Data Mining Workshops*, 2012, pp. 918-925.

[4] Guansong Pang, Shengyi Jiang, " A Generalized Cluster Centroid based classifier for text categorization", *Information Processing and Management Journal, Elsevier*, Vol. 48, No. 2, 2013, pp. 576-586.

[5] Youngjoong Ko, "A Study of Term Weighting Schemes Using Class Information for Text Classification", In *Proceedings ACM Conference on SIGIR'12*, 12-16 Aug, 2012, pp. 1029-1030.

[6] Mengen Chen, Xiaoming Jin, Dou Shen, "Short Text Classification Improved by Learning Multi-Granuality Topics", *In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, 2010, pp. 1776-1781.

[7] Aixin Sun, "Short Classification using very few words", *In Proceedings of the 35th International ACM SIGIR Conference on Research and development in Information Retrieval*, 2012, pp. 1145-1146.

[8] Svetlana Kiritchenko, Stan Matwin, "Email Classification with Co-training", *In Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative Research, 2001, CASCON '01*, pp. 8.

[9] http://en.wikipedia.org/wiki/Text_Classification

[10] M. IKONOMAKIS, S. KOTSIANTIS, V. TAMPAKAS, "Text Classification Using Machine Learning Techniques", *WSEAS Transactions on Computers*, Vol. 4, No. 8, Aug 2005, pp. 966-974.

[11]. Jiawei Han and Micheline Kamber, "Data Mining Concepts and Techniques", Second Edition, The Morgan Kauffman Series in Data Management System.