International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 01 Issue 4
MAY 2014

Influence of functional words, term weighting measures and classifiers on Text classification

Dr. P. Vijaya Pal Reddy

Professor, Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology Hyderabad
drpvijayapalreddy@gmail.com

***Abstract*— Automated text classification is a supervised learning task which uses labeled training set of documents to assign a category label to a new document based on a model generated by a classifier. The training set and test set documents needs to be preprocessed to reduce the influence of non-content words on the model derived from the training set. In this paper it is attempted to address the influence of non-content words on the classifier performance. After preprocessing the documents are represented in a machine understandable format i.e. vector space model. The terms in the document are weighted using various measures such as Term Frequency-Inverse Document Frequency (TF-IDF), Residual IDF (RIDF), x^I metric, Odds Ratio (OR(t)), Information Gain (IG(t)) Chi-squared ($\chi 2$ (t, c)) and Mutual Information (MI(t)). It is also addressed the influence of different term weighting measures on text classification in news documents. The classification model can be generated using the vector space representation of training documents set with various classifiers. In this paper an attempt is made for classification model generation using the classifiers such as Naive Bayes classifier (NB), K-Nearest Neighbor (KNN) and Support Vector Machine (SVM). The performance of the models generated using these classifiers are measured with precision, recall, F1 and macro F1 measures with various possible combinations of term weighting measures and with functional words.**

***Keywords*— Term Weighting Methods, Text Classification, Support Vector Machine, Naive Bayes, k Nearest Neighbor.**

1. INTRODUCTION

Automatic text classification (ATC) is a categorization task for labeling unlabeled documents with one of the predefined categories. ATC is a supervised machine learning technique in which the labeling to a test document can be given based on the model by learning the characteristics of each category which is specific to it. Information Retrieval (IR) and Machine Learning (ML) techniques are used to identify the keywords to represent each category specific to it. Machine learning techniques are used to design a model from which classification can be done automatically, where as Information Retrieval is used to represent the text document in a way such that it can be processed by machine learning techniques. The task of automated text classification gained a great importance in both research community and developer community since a decade as the characteristics of the text data available takes many dimensions in terms of number of words, type of words, nature of the text and language in which the text is available[1,7].

Manually classifying a large set of documents with unknown labels into predefined categories is extremely difficult task, time consuming, error prone, expensive hence not feasible. Automated text classification is a best solution for organizations where there exists a huge collection unlabeled data. Automated text categorization has reached its maturity levels with the use of full fledged techniques developed in IR and ML research. But the techniques developed in IR and ML has reached to its heights on specific languages. Hence there is a need of study on the techniques and their applicability on the language specific to the Indian context.

There are enormous applications of Text Classification in the field of science and technology. Some of them are document indexing, spam filtering, hierarchical categorization of web resources, document genre identification, automated grading of essay and categorizing news paper content into editorial, sports, politics and categorization of news paper advertisements. It is also useful in financial management, sports labeling, entertainment and medical sciences [4].

This paper deals with the comparative study of various classification techniques which are adopted from the fields of text mining, machine learning and information retrieval i.e. Naive Bayes classification based on Bayes theorem, K- Nearest Neighbor approach and Support vector machine technique comes under supervised learning techniques. In supervised learning techniques, classification models are generated using training documents which are having predefined labels [5]. Naive Bayes method calculates the probability of assigning a document to a specific category. The document can be assigned to a class with the maximum probability value. K- Nearest neighbor method calculates the distance between the test document and training documents belongs to each class. There are various measures to measure the distance. The test document is assigned to a predefined category to which it is having minimum distance.

This paper deals with classification methods starts with the preprocessed documents and then significant terms are obtained from the training documents after preprocessing. The significance of a term specific to a document, to a category and to a corpus is measured using various term weighting measures. These weighted terms are used to train the classifier. After training phase is completed it results to a generation of set rules known as classification model to predict the category label of unknown label documents. The effect of term weighting measures, the effect of functional words in deriving the classification rules and the efficiency of classifiers are measured using precision, recall and F1 measures for a set of test documents.

Extensive research works for text classification have been not yet conducted on Telugu text documents since Telugu language is highly rich in its morphology and requires special treatments such as ordering verbs, morphological analysis, etc. In Telugu morphology, words have affluent meanings and contain a great deal of grammatical and lexical information. Telugu text documents are required significant processing to build accurate classification model. In this work, single label binary categorization on labeled training data is carried out on Telugu language text using various term weighting approaches. As in [6,16] the experiments have been conducted on Telugu text using different term weighting schemes such as Term frequency, TF-IDF, TF-RF, TF- CHI-SQUARED with an assumption that the term frequency is a good factor for discriminating the categories among each other, but in this paper by keeping view of document length as a primary concern and the term frequency in not an influencing factor for term weighing. Hence other term weighting schemes are applied on Telugu text documents for text classification.

This paper is organized into five different sections. Chapter 1 gives us the introduction and brief explanation about classification. Chapter 2 deals with the explanation of classification model, preprocessing, term weighting approaches and various classifiers. We discuss language characteristics in Chapter 3. All experimental results obtained are tabulated, compared and evaluated in Chapter 4. In Chapter 5, we conclude our paper with a brief overview of future work.

2. TEXT CLASSIFICATION PROBLEM

Classification is the task of categorization of data to utilize for various applications. It is a supervised technique in which classification of unknown documents can be done by deriving a set of rules from training samples whose class labels are known in advance. It is a technique of data mining. Let (dj, ci) Є D >> C, where D is a set of documents and C= {c1, c2....c|C|} are set of predefined categories. The task of Text Classification is to assign a Boolean value to each pair in D [9].

The of task of classification is to construct a model to predict labels for test documents which are represented in vector space format. The feature vectors are formed for each document in the corpus by measuring each term with a weight. The weight given to a term represents the importance of a term in a document or in a category. Classification can be divided into three phases. They are 1) feature vector generation with appropriate weights to each term 2) classifier training with training set feature vectors and 3) Derive a set of classification rules from classifier to predict class label.

In the first phase, the text documents are represented in the form feature vectors. The feature vectors are machine understandable way of representation of documents to process by the machine. The document is mapped into a feature space and each feature is measured with a weight in feature space. To reduce the influence of the functional words on the feature vector space, the documents are preprocessed as shown in Figure 1. The preprocessing phase consists of various steps such as tokenization, stop word removal and stemming. After document preprocessing, each feature in the document are represented with a particular weight such that the most

important features are more emphasized and less important features are either removed or their importance in the document is reduced with low weights.

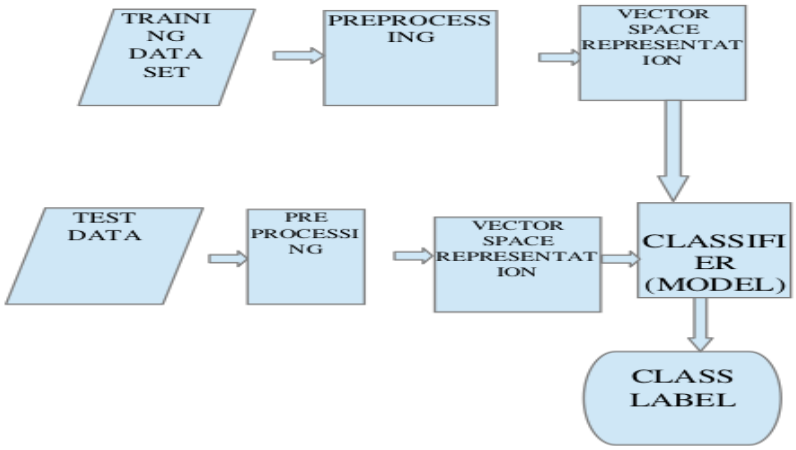

FIGURE 1: TEXT CLASSIFICATION MODEL

The second phase is used to train the classifier with the feature vectors of the training samples with their labels. As an output of the classification from the classifier produces a model with a set of classification rules. The model created from the classifier differs from each classifier. The accuracy of the generated model can be measured using test documents whose class labels are known in advance. The machine generated class labels for test documents are compared with the predefined class labels to estimate the accuracy of the classifier.

In the third phase, after improving the accuracy of the classifier to a satisfactory level, it can be used to predict the labels of unknown documents whose labels are not known in advance. In this phase an unlabeled document are taken as input to the model and produces a label of the inputted document as an output. The various steps in the preprocessing phase are as follows:

A. TOKENIZATION

Articles in the news papers follows their own font style. To achieve the uniformity, the documents are converted into Unicode format. The Unicode formatted documents are divided into a set of terms called tokens. The symbols such as punctuations, question marks, exclamatory marks and other symbols which are not useful for classification are removed from the token list of each document. The term may be a word, a set of words, combination of syllables in the words with various lengths
[17] termed as n-grams. This process is called Tokenization.

A. STOP WORD REMOVAL

A stop word list is a list of commonly repeated features which occur in most of the documents. The common features such as pronouns, conjunctions and prepositions need to be removed because these stop words does not help to decide the category of the document. For the same reason, if the feature is a special character or a number then that feature should be removed. Stop words are removed from all the documents in order to reduce the dimensionality of the training set so that curse of dimensionality problem can be addressed. It also reduces the time to produce the classification model. Stop word list is identified using Natural Language tool kit (NLTK) called Telugu tagger. The Telugu tagger is trained on a tagger named as telugu.pos from the Indian corpus that comes with NLTK.

C. STEMMING

Stemming is a process of transforming the terms into their root form or basic form. This process helps in improving into their root form or basic form. This process helps in improving the performance of the system by mapping all variants of same word into a single word. The term frequency of the stemmed word becomes the cumulative frequency of all the words belongs to the same basic form. This process reduces the space and time complexity of the system. By using the tool Telugu morphological analyzer (TMA) stem forms of the inflected words are identified.

D. VECTOR SPACE MODEL

Vector space model [2] is used to represent the document in computer understandable form.In this model each document is represented as a vector in a vector space. Each dimension in the vector is the weight of each feature. In this paper, it is attempted to address the influence of term weight on classification. Each document can be represented as a vector of the form d =(t_1, w_1;t_2, w_2;.... ;t_n,w_n), where t_i is a term, w_i is the weight of the term t_i in the document d. A term may be a word, a phrase or n- grams. The most appropriate term weighting scheme gives larger weight to most useful terms for classification and less weight for the other terms. There is different term weighting methods proposed in the text classification which are studied for Telugu text classification in this paper. In this paper, we considered the seven term weighting approaches which are proved to be prominent in the text classification for Telugu news articles with short length. The various term weighting schemes are as follows:

1) *Term Frequency-Inverse Document Frequency (TF- IDF)*

Term frequency (TF) is the number of times a term appears within a document. Inverse Document Frequency (IDF) is the frequency of a term that occur in other documents. TF-IDF gives larger value to a term which occurs few times in other documents within the corpus and that occur many times within a document. TF is a good statistic measure to identify the importance of a word. If TF of a word is more then it could be important. This is the most common term weighting method as in [15] and it is used in information retrieval. The idea behind TF- IDF is to find the terms that occur most often within the document (TF) and occur rarely in other documents (IDF):

$$TF-IDF(t,d)=-\log\left(\frac{df(t)}{N}\right)*\frac{tf(t,d)}{(d)} \quad (1)$$

Where tf(t, d) is the term frequency of word t in

the document d, |d| is the number of words in the document, df(t) is the number of documents with at least one occurrence of t and N is the number of documents in the corpus.

2) *Residual IDF (RIDF)*

Residual IDF as in [9] is based on the idea of comparing the word"s observed IDF with the predicted IDF (PIDF). Predicted IDF is calculated using the term frequency by assuming a random distribution of the term in the documents. The higher the difference between IDF and PIDF means the word is more informative.

$$RIDF(t)=-\log\left(\frac{df(t)}{N}\right)+\log\left(1-e^{\frac{-ctf(t)}{N}}\right) \quad (2)$$

Where ctf(t) is the collection term frequency. This approach gives highest weight to the words with medium frequency.

3) *x^I metric*

$$x^I(t)=N_t-df(t) \quad (3)$$

x^l metric is introduced by Book stein and Swanson [11]. Where N_t is the total number of occurrences of t, and df (t) is the number of documents where t exists.

4) *Odds Ratio (OR (t))*

Odds Ratio (OR (t)) is for relevance ranking in information retrieval [12]. It is calculated by taking the ratio of positive samples and negative samples [12]:

$$ODDSRATIO(t) = \log\left(\frac{N_{t,c} \times N_{\neg t,\neg c}}{N_{t,\neg c} \times N_{\neg t,c}}\right) \quad (4)$$

where $N_{t,c}$ denotes the number of times term t occurs in category c, $N_{t,\neg c}$ is the number of times t occurs in other categories than c, $N_{\neg t,\ c}$ is the number of times c occurs without term t, $N_{\neg t,\neg c}$ is the number of times neither c nor t occurs.

5) *Information Gain (IG (t))*

Information Gain (IG(t)) measures the entropy of a feature with its presence and its absence. This is the difference in observed entropy H(C) and the expected entropy ET(H(C|T)) [13]:

$$P(\neg t)\sum_{i=1}^{m} P(c_i/\neg t)\log P(c_i/\neg t) \quad (5)$$

where ¬t indicates the absence of t, m is the number of categories, c_i is the i^{th} category.

6) *Chi-squared (χ2 (t,c))*

Chi-squared (χ2 (t, c)) is a statistical test. In feature weighting it is used to assess the dependency of the feature category or feature and feature pairs [15]:

$$X^2(t,c) = \frac{N \times (A \times D - C \times B)^2}{(A+C) \times (A+B) \times (B+D) \times (C+D)} \quad (6)$$

where A = $N_{t,c}$, B = $N_{t,\neg c}$, C = $N_{\neg t,c}$, and D = $N_{\neg t,\neg c}$. If the χ2 score is large then the feature is important for the category.

7) *Mutual Information (MI(t,c))*

Mutual Information (MI(t,c)) is to score each feature and category pair or feature and feature pair to measure feature contributes to the pair:

$$MI(t,c) = \log_2\left(\frac{Nt,c \times N}{(Nt,c+N\neg t,c) \times (Nt,c+Nt,\neg c)}\right) \quad (7)$$

E. CLASSIFIERS

Classifiers are used to generate a model from the training set vector space. The derived models varies

based on the selected classification approach. There are many classification methods are developed in the machine learning such as Bayesian classification model (NB), K-nearest neighbor classification (KNN), decision trees induction (DT), Support vector machines (SVM), back propagation (BP) and Neural Networks (NN). Support Vector Machines (SVM) has the ability to efficiently handle relatively high dimensional and large-scale data sets without decreasing classification accuracy. K-nearest neighbor (kNN) approach is based on the k training documents which are closest to the test document. It is very simple and effective but not efficient in the case of high dimensional and large-scale data sets. The Naive Bayes (NB) method assumes that the terms in a document are independent to each other which is not the case in the real world scenario. In this paper, considered the classifiers such as NB, KNN and SVM for classification of Telugu text classification. The brief descriptions about these approaches are given below:

1) Naive Bayes Algorithm

Naive Bayes classifier is one of the simplest probabilistic Bayesian categorization approach. The assumption in NB classifier is that the effect of an attribute value on a given category is independent of the values of other attributes known as conditional independence. It is used to simplify complex computations [14]. The Naive Bayes classifier is based on the Naïve bayes assumption. From Bayes rule, the posterior probability is calculated as

$$P(c/x) = \frac{P(c)P(x/c)}{P(x)}$$

where x is a feature vector space of a document and x =(x 1 ,...,x n) and c is category. The parameter P(c) is estimated as

$$P(c) = \frac{NumberofdocumentsofC}{TotalNumberofdocuments}$$

The categorization results are not affected because parameter p(x) is independent of categories. Assuming that the components of feature vectors are statistically independent of each other. P(x | c) can be calculated as

$$P(x/c) = \prod_{i}^{m} P(x_i/c)$$

The Naive Bayes classifier predicts the category C max with the largest posterior probability [11]:

$$C_{max} = argmaxc \; P(c/x)$$
$$= argmaxc \; P(c)P(x/c)$$

2) KNN Algorithm

The k-Nearest Neighbor (k-NN) categorization is the simplest among all the supervised machine learning techniques but widely used method for classification and retrieval. It is an instance based learning and often called lazy learning algorithm. A k-Nearest Neighbor classifier (kNN) is used to find the k nearest training vectors for the given test vector and use these categories from those k vectors as the category labels for the test vector. The distance between the test vector and the training vector can be calculated in

many ways such as number of matching features, a cosine similarity between the feature vectors and Euclidean distance between the feature vectors. The label can be selected from the closest k neighbors. Other possibility is the assigning the label that occurs most frequently among the k neighbors or use all of the labels among the neighbors in the ranking order as in [8].

In this paper, the Euclidean distance is used as a measure to find the distance between two vectors. Nearest Neighbor algorithm is a particular instance of k-NN where k=1. The formula is defined as follows as in [9].

$$d(\mathbf{x}_i, \mathbf{x}_l) = \sqrt{(x_{i1} - x_{l1})^2 + (x_{i2} - x_{l2})^2 + \cdots + (x_{ip} - x_{lp})^2}.$$

3) Support Vector Machines (SVM)

Support Vector Machine classifier (SVM) [10] takes the training vectors and aims to find a hyper plane that separates positive and negative samples into different sides of the hyper plane. Usually it is impossible to separate the samples directly using the given data in the given dimensions. For this reason, it is often a good idea to map the original space into a higher-dimensional space where the separation is easier to accomplish [11]. SVM classifiers use a kernel function that maps the features into higher dimensions and creates the hyper plane; this is the model created by the simple SVM classifier.

3. TELUGU LANGUAGE CHARACTERISTICS

There are more than 150 different languages spoken in India today. Many of the languages have not yet been studied in any great detail in terms of Text Categorization. 22 major languages have been given constitutional recognition by the government of India. Modern Indian languages are characterized by a rich system of inflectional morphology and a productive system of derivation. This means that the number of surface words will be very large and so will be the raw feature space, leading to data sparsity. Dravidian morphology is in particular more complex. Dravidian languages such as Telugu and Kannada are morphologically among the most complex languages in the world, comparable only to languages like Finnish and Turkish. The main reason for richness in morphology of Telugu (and other Dravidian languages) is, a significant part of grammar that is handled by syntax in English (and other similar languages) is handled within morphology. Phrases including several words in English would be mapped on to a single word in Telugu. Hence there is a necessity to study the influence of term weighting methods on different classification approaches on Indian context.

1) Test collections

The dataset was gathered from Telugu News Papers such as Eenadu, Andhra Prabha and Sakshi from the web during the year 2009 – 2010. The corpus is collected from the website http://uni.medhas.org/ in Unicode format. We obtained around 800 news articles from the domains of economics, politics, science, sports, culture and health. Before proceeding, we conduct some preprocessing like tokenization, removing stopping words and stemming. Considered 70% of the documents as training samples, remaining 30% of the documents as testing samples for all six categories. We conducted the experiments using TF- IDF, RIDF, XI METRIC, ODDS RATIO, INFORMATION GAIN, CHI-SQUARED and MUTUAL INFORMATION in combination with Naive Bayes, KNN and SVM classifiers. After the experiment, we compare result of different weighting methods with three classifiers without preprocessing the documents and after preprocessing the documents.

2) Evaluation Methods

In order to compare the results of all possible combinations of term weighting methods with classifiers before and after preprocessing computed the precision, recall, F1 measure and macro-averaged F1 measure . Precision is the number of correct categories out of all the predicted categories and recall is the number of correct categories out of all the categories of the document, where F1 is the standard measure for test accuracy used in several research articles in this area [14, 15] and is computed based on the following equation:

$$F_1 = \frac{2 * Recall * Precision}{Recall + Precision}$$

where,

$$Precision = \frac{X}{X+Y}$$

$$Recall = \frac{X}{X+Z}$$

where X is documents retrieved relevant, Y is documents retrieved irrelevant and Z is documents not retrieved but relevant. Macro-averaged F-Measure is computed locally over each category first and then the average over all categories is taken. Macro-averaged F- measure is obtained by taking the average of F-measure values for each category as:

$$F(macro-average) = \frac{\sum_{1}^{M} F_i}{M}$$

where M is total number of categories. Macro-averaged F-measure gives equal weight to each category, regardless of its frequency.

We have used the linear SVM for SVM classification [6] and for KNN classifier, k value as 1. In KNN algorithm, we have used the Euclidian distance measure as a similarity measure to find the distance between training document and text document. The corpus detains are shown in Table: 1 and the experimental results are shown in Table: 2, 3, 4, 5, 6 and 7 for F1 and Macro averaged F1 results of NB Classifier for six categories, F1 and Macro averaged F1 results of KNN Classifier for six categories,F1 and Macro averaged F1 results of SVM Classifier for six categories respectively.

CATEGORY	NO. OF TRAINING DOCUMENTS	NO.OF TESTING DOCUMENTS	TOTAL NO. OF DOCUMENTS
Economics	60	40	100
Politics	120	80	200
Science	90	60	150
Sports	75	48	123
Culture	54	36	90
Health	85	50	135

Table 1: Corpus statistics

Category	TF-IDF	RIDF	XI	OR	IG	CHI^2	MI
Economics	0.635	0.618	0.683	0.724	0.690	0.729	0.719
Politics	0.701	0.659	0.718	0.748	0.724	0.730	0.726
Science	0.719	0.698	0.725	0.752	0.731	0.741	0.719
Sports	0.648	0.653	0.717	0.775	0.709	0.757	0.737
Culture	0.714	0.695	0.729	0.752	0.731	0.738	0.749
Health	0.695	0.707	0.720	0.757	0.717	0.741	0.738
F(macro-averaged)	0.685	0.671	0.715	0.751	0.717	0.739	0.731

Table 2:F1 and Macro averaged F1 results of NB Classifier for six categories before preprocessing

Category	TF-IDF	RIDF	XI	OR	IG	CHI^2	MI
Economics	0.737	0.723	0.786	0.825	0.787	0.791	0.809
Politics	0.791	0.743	0.798	0.837	0.781	0.828	0.806
Science	0.793	0.763	0.785	0.819	0.813	0.812	0.805
Sports	0.757	0.748	0.803	0.845	0.808	0.827	0.818
Culture	0.806	0.768	0.791	0.829	0.789	0.818	0.799
Health	0.789	0.802	0.815	0.853	0.797	0.847	0.828
F_MG	0.778	0.757	0.796	0.834	0.795	0.820	0.810

Table 3:F1 and Macro averaged F1 results of NB Classifier for six categories after preprocessing

Category	TF-IDF	RIDF	XI	OR	IG	CHI^2	MI
Economics	0.651	0.623	0.721	0.735	0.718	0.720	0.729
Politics	0.678	0.648	0.723	0.769	0.720	0.746	0.738
Science	0.680	0.686	0.740	0.743	0.709	0.731	0.729
Sports	0.717	0.689	0.707	0.782	0.717	0.767	0.748
Culture	0.705	0.703	0.738	0.768	0.742	0.754	0.761
Health	0.725	0.713	0.743	0.750	0.708	0.739	0.745
F_MG	0.692	0.677	0.728	0.757	0.719	0.742	0.741

Table 4: F1 and Macro averaged F1 results of KNN Classifier for six categories before preprocessing

Category	TF-IDF	RIDF	XI	OR	IG	CHI^2	MI
Economics	0.748	0.730	0.751	0.845	0.801	0.812	0.826
Politics	0.736	0.739	0.743	0.832	0.784	0.831	0.814
Science	0.761	0.758	0.780	0.809	0.807	0.809	0.822
Sports	0.772	0.760	0.814	0.867	0.821	0.831	0.824
Culture	0.816	0.771	0.818	0.835	0.812	0.823	0.806
Health	0.812	0.798	0.826	0.851	0.805	0.839	0.819
F-MG	0.774	0.759	0.788	0.839	0.805	0.824	0.818

Table 5: F1 and Macro averaged F1 results of KNN Classifier for six categories after preprocessing

Category	TF-IDF	RIDF	XI	OR	IG	CHI^2	MI
Economics	0.685	0.672	0.699	0.753	0.725	0.728	0.737
Politics	0.690	0.685	0.702	0.763	0.736	0.752	0.742
Science	0.723	0.717	0.732	0.756	0.715	0.731	0.738
Sports	0.720	0.705	0.727	0.779	0.722	0.770	0.731
Culture	0.726	0.712	0.746	0.797	0.751	0.768	0.779
Health	0.739	0.724	0.752	0.789	0.749	0.774	0.764
F-MG	0.713	0.702	0.726	0.772	0.733	0.753	0.748

Table 6: F1 and Macro averaged F1 results of SVM Classifier for six categories before preprocessing

Category	TF-IDF	RIDF	XI	OR	IG	CHI^2	MI
Economics	0.772	0.758	0.781	0.875	0.824	0.838	0.853
Politics	0.761	0.765	0.770	0.862	0.803	0.858	0.832
Science	0.787	0.783	0.803	0.838	0.831	0.831	0.831
Sports	0.793	0.785	0.834	0.884	0.846	0.849	0.853
Culture	0.841	0.794	0.841	0.862	0.838	0.849	0.835
Health	0.836	0.821	0.852	0.876	0.831	0.868	0.848
F-MG	0.798	0.784	0.813	0.866	0.828	0.848	0.842

Table 7: F1 and Macro averaged F1 results of SVM Classifier for six categories after preprocessing

V. RESULTS AND DISCUSSIONS

We compared the performance of the term weighting approaches such as chi-squared, Mutual Information, Information Gain, Odds Ratio, Residual IDF, TF-IDF, and RIDF. From Table 2 to Table 7 shows the results of the evaluation. The results of Mutual Information are comparable with the results of Information Gain. When the size of the document is short then the impact of term weighting is also very less as there only less features for classifier to generate a model. This is reason the there is only small increments in the F1 measure from SVM classifier to other classifiers such as Naive Bayes classifier and K-Nearest Neighbor classifier. If the documents contain more number of features then the impact of the classifier is more visible from the text classification.

From the results, it can be observed that the SVM classifier is the best performer when compared with Naive Bayes classifier and K-nearest neighbor classifier in both the cases such as before preprocessing and after preprocessing for most of the data sets. The performance is measured using F1 score and F-macro average score. ODDS RATIO is performing better compared with other weighing measures with the combination of Support vector machine. Best macro averaged-F is achieved by using the ODDS RATIO scheme. Mutual information and chi- squared are also performing well when compared with other term weighting measures. It is observed that the positive samples and negative samples place a vital role in discrimination when compared with term frequency and inverse document frequency. The TF_IDF, RIDF weighting schemes and x I metric are low performers when compared with the remaining schemes in most of the categories in all classifiers. It shows that TF, IDF and DF are not much influencing factors for text classification.

Moreover for the Telugu data sets, the SVM classifiers have higher macro-averaged F1 than NB, KNN respectively for both te cases of before preprocessing and after preprocessing. Another notable result that

was also reported is that all classifiers vary among categories. The "Culture" category has a neat classification with F1 of 79.7%, while the “Economics” category has a noticeably poor F1 measure of 75.3% for SVM before preprocessing. The "Sport" category has a neat classification with F1 of 88.4%, while the “science” category has a noticeably poor F1 measure of 83.8% for SVM after preprocessing. These poor results indicate that the "Science" category is highly overlapped with other categories.

V. CONCLUSIONS AND FUTURE SCOPE

The text classification can be performed in three steps:1) Preprocessing and then the features are weighted with various features to build the feature vectors space. 2) The classifier to generate model for test set classification. This step produces set of classification rules. 3) The test documents, which do not have a label, are classified using the trained classifier. These experiments were made by dividing the data into training set and the test set. The macro average F1 of seven term weighting measures were obtained against six Telugu category sets indicated that the SVM algorithms dominant NB and KNN algorithms. Finally, SVM and KNN classifiers perform excellent in most of the categories. ODDS RATIO scheme shown good performance compared with other six variants of term weighting schemes. The TF_IDF, RIDF and XI metrics do not improve the term‟s discriminating power for text categorization. With this empirical analysis we are planning to use ODDS RATIO as the term weighing scheme for further research on Telugu Text categorization. Also, planning to propose a hybrid approach, a combination of text summarization with text classification to increase the accuracy of the text classification process on Telugu documents.

REFERENCES

[1] Arturo Montejo-Raez, Thesis on „Automated Text Categorization of documents in the High Energy Physics domain.

[2] Gerard Salton, A. Wong, C. S. Yang, “A Vector Space Model for Automatic Indexing”, CACM 18(11), 1975.

[3] Fabrizio Sebastiani, ' Text Categorization', University of Padova, Italy, 2005.

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

[5] Vishnu murthy et al. “ A COMPARATIVE STUDY ON TERM WEIGHTING METHODS FOR AUTOMATED TELUGU TEXT CATEGORIZATION WITH EFFECTIVE CLASSIFIERS “,International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.3, No.6, November 2013 [6] Fabrizio Sebastiani, 'Machine Learning in Automated Text Categorization',Italy2002.

[7] K.W. Clark and W.A. Gale. Inverse Document Frequency (IDF): A measure of deviation from Poisson. In Third Workshop on Very Large Corpora, Massachusetts Institute of Technology Cambridge, USA, pages 121–130, June 1995.

[8] A. Bookstein and D. R. Swanson. Probabilistic models for automatic indexing. Journal of the American Society for Information Science, 5(25):312–318, 1974.

[9] D. Mladenic and M. Grobelnik. Feature selection for unbalanced class distribution and Naive Bayes. In Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Slovenia, pages 258–267, June 1999.

[10] Thesis on „Clustering Approaches Categorization‟ by Hiroya Takamura to Text

[11] C. Cortes and V. Vapnik. 20(3):273–297, 1995. Support-vector networks. Machine Learning,

[12] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. Numerical Recipes: The Art of Scientific Computing (3rd ed.), chapter 16.5. Support Vector Machines. New York: Cambridge University Press, 2007.

[13] G. Forman. An extensive empirical study of feature selection metrics for text classification. Journal of Machine Learning Research, 3:1289–1305, 2003.

[14] G. Forman. BNS feature scaling: an improved representation over TF-IDF for SVM text classification. In Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM‟08), USA, pages 263–270, October 2008.

[15] T. Joachims. Text categorization with Support Vector Machines: Learning with many relevant features. In 10th European Conference on Machine Learning (ECML‟98), Germany, pages 137–142, April 1998.

[16] Y. Yang. An evaluation of statistical approaches to text categorization. Information Retrieval, 1(1-2):69–90, 1999.

[17] Y. Yang and J.P. Pedersen. Feature selection in statistical learning of text categorization. In Proceedings of the Fourteenth International Conference on Machine Learning (ICML‟97), USA, pages 412–420, July 1997.