A COMPARATIVE STUDY ON TERM WEIGHTING METHODS FOR AUTOMATED TELUGU TEXT CATEGORIZATION WITH EFFECTIVE CLASSIFIERS

Vishnu Murthy.G[1], Dr. B. Vishnu Vardhan[2], K. Sarangam[3] and
P. Vijay pal Reddy[4]

[1]Department of Computer Science and Engineering, AGI, Hyderabad
[2]Department of Computer Science and Engineering, JNTU, Jagityal
[3]Department of Computer Science and Engineering, TEC, Hyderabad
[4]Department of Computer Science and Engineering, MRCE, Hyderabad

ABSTRACT

Automatic Text categorization refers to the process of assigning a category or some categories automatically among predefined ones. Text categorization is challenging in Indian languages has rich in morphology, a large number of word forms and large feature spaces. This paper investigates the performance of different classification approaches using different term weighting approaches in order to decide the most applicable one to Telugu text classification problem. We have investigated on different term weighting methods for Telugu corpus in combination with Naive Bayes (NB), Support Vector Machine (SVM) and k Nearest Neighbor (kNN) classifiers.

KEYWORDS

Term Weighting Methods, Text Categorization, Support Vector Machine, Naive Bayes, k Nearest Neighbor, CHI-square.

1. INTRODUCTION

Now a days it is a challenge that the Information retrieval from the available large amount of document data, where the information is accessed exactly required by the used and quickly. Information Retrieval (IR) is searching for information. Information can be retrieved from relational databases, documents, text, multimedia files, and the World Wide Web [9]. The applications of IR are extraction of information from large documents, searching in digital libraries, information filtering, spam filtering, object extraction from images, automatic summarization, document classification and clustering, and web searching.

DOI : 10.5121/ijdkp.2013.3606

Text Categorization (TC) is also known as text classification. It is the task of automatically sorting a set of documents into categories (or topics, classes) from a predefined set. Automated text classification tools are available and used to organize the documents into classification.

There are two types of categorization. First is Rule-based and the second is Machine Learning [4]. In Rule-based approach classification rules are framed manually and the documents classified based on rules. In Machine learning approaches equations are defined automatically using sample labeled documents.

TC involves many applications such as Automatic indexing for Boolean information retrieval systems,Text filtering, Word sense disambiguation, Hierarchical categorization of Web pages, identification of document genre, authorship attribution [14].

Extensive research works have not been conducted on Telugu corpus since Telugu language is highly rich and requires special treatments such as ordering of 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 in order to build accurate classification model. In this work, single label binary categorization on labeled training data is carried out on Telugu language text. So far no comparisons were made against Telugu language data collections for different term weighting methods with various classification algorithms.

The rest of the paper is organized as follows: Text Categorization model, Preprocessing with reference to Telugu text corpus, different term weighting approaches and classification approaches are explained in Section 2. Section 3 describes the characteristics of Telugu language. Section 4 is dealt with data collection as well as the experimentation. Section 5 is with results analysis, and finally the conclusion with further research is given in Section 6.

2. TEXT CATEGORIZATION PROBLEM

Text categorization is the task of assigning test documents into predefined categories. Let 'D' is a document domain and C = {c_1 , c_2 , ..., $c_{|c|}$ } is a set of predefined categories. Then the task is, for each document dj Є D, a decision to assign document d_j under c_i or a decision not to assign d_j under c_i (ci Є C) by virtue of a function Φ, where the function Φ is also called the classifier [13].

The TC problem can be modeled as shown in Figure.1. The proposed system having three modules mainly such as text document preprocessing, classifier construction and performance evaluation. Document collection is divided into two sets: Training set and Test set. Training set is a pre-classified set of documents which are used for training the classifier, while the Testing set is to determine the accuracy of the classifier whether the set is having correct and incorrect classifications for each input. The different phases in the model are explained below.

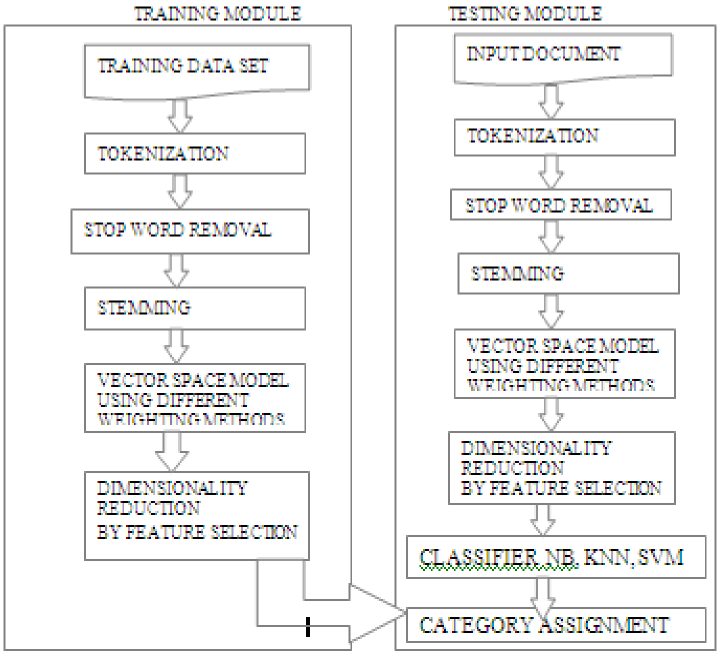
TRAINING MODULE
TRAINING DATA SET
TOKENIZATION
STOP WORD REMOVAL
STEMMING
VECTOR SPACE MODEL USING DIFFERENT WEIGHTING METHODS
DIMENSIONALITY REDUCTION BY FEATURE SELECTION
TESTING MODULE
INPUT DOCUMENT
TOKENIZATION
STOP WORD REMOVAL
STEMMING
VECTOR SPACE MODEL USING DIFFERENT WEIGHTING METHODS
DIMENSIONALITY REDUCTION BY FEATURE SELECTION
CLASSIFIER NB, KNN, SVM
CATEGORY ASSIGNMENT

FIGURE 1: TEXT CATEGORIZATION MODEL

A. TOKENIZATION

A document is divided into small units called tokens. This process is called Tokenization. This results a set of atomic words with semantic meaning [11]. This phase outputs the article as a set of words by removing the unnecessary symbols like semicolons, colons, exclamation marks, hyphens, bullets, parenthesis, numbers etc.

B. STOP WORD REMOVAL

A stop list is a list of commonly repeated features such as pronouns, conjunctions and prepositions etc. which appear in every text document. These features are to be removed because they do not have effect on the categorization process. For example, if the feature has a special character or a number then the feature is removed. 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. The accuracy is almost 98%.

C. STEMMING

By removing affixes, prefixes and/or suffixes from features is known as Stemming. It is used to reduce the number of features in the feature space and improve the performance of the classifier when the different forms of features are stemmed into a single feature. By using the tool Telugu Morphological Analyzer (TMA) developed by IIT, Hyderabad and Central university of Hyderabad, stem forms of the inflected words are identified.

D. VECTOR SPACE MODEL

The basic idea of vector space model [2] is representing the document in computer understandable form. Bag-of-word model is one of the forms to represent the document followed in this paper. In Space Vector Model, any text document is represented as vectors or dimensions. Each dimension of space is represented as a single feature of the vector and the weight is calculated by various weighting schemes. Hence, each document can be represented as d =(t_1, w_1;t_2, w_2;.... ;t_n,w_n), which t_i is a term, w_i is the weight of the t_i in the document d. in order to reflect the importance of the term in a document we use term weighting. There is different term weighting methods proposed in the TC study. In this paper, we considered the four term weighting approaches which are proved to be prominent in TC for Telugu Text categorization. They are as defined as follows:

1) Term Frequency (TF)

Using this method [10][11], each term is assumed to have a value proportional to the number of times it occurs in a document is as follows:

$$W(d,t) = TF(d,t)$$

2) Term Frequency-Inverse Document Frequency (TF-IDF)

This approach follows Salton's definition [4][5], which combined TF and IDF to weight the terms and the author showed that this approach gives better performance with reference to accuracy that IDF and TF alone. The combined result of TF and IDF is given as:

$$W(d,t) = TF(d,t).IDF(t)$$

and for a given N documents, if n documents contain the term t, IDF is given as follows:

$$IDF(t) = log\left(\frac{N}{n}\right)$$

3) Term Frequency-Chi square (TF.CHI)

The TF.CHI scheme [12] is included for two reasons. First, it is a typical representation which combines TF factor with one feature selection metric i.e. CHI-square.

4) Term Frequency-Relevance Frequency (TF.RF)

This term weighting method in proposed in [17].According to the proposal, this is the best term weighting approach for TC on English documents. Hence this method is considered to study on Telugu TC. This approach is defined as follows:

$$TF.RF(t) = TF * log\left(2 + \frac{a}{max(1,c)}\right)$$

where, a is the number of documents which contain the positive category term, c is the number of documents which contain the negative category term.

E. DIMENSIONALITY REDUCTION

The feature space is too large even after removing non-informative features and the stemming process. Features which are not positively influencing the TC process can be removed without affecting the classifier performance, known as Dimensionality reduction (DR). DR of the feature space is carried out by feature selection and feature extraction.

Feature selection deals with several methods such as document frequency, DIA association factor, chi-square, information gain, mutual information, odds ratio, relevancy score, GSS coefficient. These methods are applied to reduce the size of the full feature set. DR by feature extraction is used to create a small set of artificial features from original set. This feature can be computed by using Term clustering and Latent semantic indexing features.

In Indian languages, the number of features are even higher compared with English text because of richness in morphology. We use $\chi2$ metric [1] for feature selection in this paper, which are found $\chi2$ and information gain are the most effective feature selection metrics in the literature. CHI square measures the correlation between feature and class.

For example; Let A be the times both feature t and class c exists, B be the times feature t exists, but class c doesn't exist, C be the times feature t doesn't exist, but class c exists, D be the times both feature t and class c doesn't exist, N be the total number of the training samples. Then CHI square (χ^2) statistics can be written as:

$$X^2(t,c) = \frac{N*(AD-BC)^2}{(A+C)*(B+D)*(A+B)*(C+D)}$$

F. CLASSIFIERS

There are many classification approaches such as Bayesian model, decision trees, Support vector machines, Neural Networks and K- nearest neighbor, etc.. in the literature for Text Categorization. Support Vector Machines (SVM) has a better performance than other methods due to its ability to efficiently handle relatively high dimensional and large-scale data sets without decreasing classification accuracy. K-nearest neighbor (kNN) makes prediction 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 one document are independent even this is not the case in the real world. In this paper, we considered the SVM,KNN and NB classification approaches for Telugu Text Categorization. The brief description about these methods are given below:

1) Naïve Bayes Algorithm

Naive Bayes [8] is a supervised, probabilistic learning method and is computed as:

$$P(d\,|\,c) = \prod_{1 \le i \le nd} P(w_i\,|\,c).$$

where $P(w_i|c)$ is the conditional probability of term w_i occurring in a document of class c. We interpret $P(w_i|c)$ as a measure of how much evidence w_i contributes that c is the correct class. $(w_1, w_2, . . ., w_n)$ are the tokens in the document 'd'; are part of the vocabulary used for classification and 'n' is the number of such tokens in the document d. In text classification, the maximum a best class in Naive Bayes classification is the most likely or maximum a posteriori and denoted by:

$$c_{\text{map}} = \underset{c \in C}{\arg\max} \prod_{1 \le i \le nd} P(w_i \mid c)$$

2) KNN Algorithm

K-nearest neighbor (KNN) which is also known as Text-to-Text Comparison (TTC), is a statistical approach, which is has been successfully applied to TC problem [13] and showed promising results. Given a test document to be classified, the algorithm searches for the K nearest neighbors among the pre-classified training documents based on some similarity measure and ranks those k- neighbors based on their similarity scores, the categories of the k-nearest neighbors are used to predict the category of the test document by using the ranked scores of each as the weight of the candidate categories, if more than one neighbor belong to the same category then the sum of their scores is used as the weight of that category, the category with the highest score is assigned to the test document provided that it exceeds a predefined threshold, more than one category can be assigned to the test document.

3) Support Vector Machines (SVM)

In general, SVM[7] is a linear learning system that builds two-class classifiers. Let the set of training examples D be {(x1, y1), (x2, y2), ..., (xn, yn)}, where xi = (xi1, xi2, ..., xir) is a r-dimensional input vector in a real-valued space, yi is its class label (output value) and yi belongs to {1, -1}. 1 denotes the positive class and -1 denotes the negative class. To build a classifier, SVM finds a linear function of the form:

$$f(x) = w.x + b$$

so that an input vector xi is assigned to the positive class if f(xi) >= 0, and to the negative class otherwise.

$$y_i = \begin{cases} 1 & \text{if} \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \ge 0 \\ -1 & \text{if} \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b < 0 \end{cases}$$

Hence, f(x) is a real-valued function, w = (w1, w2, ..., wr) is the weight vector, b is called the bias, <w . x> is the dot product of w and x. Without using vector notation, Equation can be written as:

$$f(x_1, x_2, \ldots\ldots x_r) = w_1 x_1 + w_2 x_2 + \ldots\ldots + w_r x_r + b$$

3. TELUGU LANGUAGE CHARACTERISITCS

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.

Indian languages are characterized by rich system 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. The small phrases in English are mapped on to a single term in Telugu language. Hence there is a necessity to study the influence of term weighting methods on different classification approaches on Indian context.

4. Emperical Evaluation

1) Test Collections

The dataset was gathered from Telugu News Papers such as Eeenadu, 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 tokenisation, removing stopping words and stemming choose 60% of the documents as training samples, remaining 40% of the documents as testing samples for all six categories. Then we use CHI square statistics feature selection method to select 100 features, and then we conduct the experiments using TF, TF-IDF, TF-CHI and TF-RF weighing methods separately on classifiers such as Naïve Bayes, KNN and SVM. After the experiment, we compare result of different weighting methods with three classifiers.

2) Evaluation Methods

In order to compare the results of all possible combinations of term weighting methods with classifiers, we computed the precision, recall, F1 measure and macro-averaged F1 measure . Precision is the proportion of examples labeled positive by the system that were truly positive, and recall is the proportion of truly positive examples that were labeled positive by the system. where F1 is computed based on the following equation:

$$F_1 = \frac{2*Recall*Precision}{Recall+Precision} \quad \text{where,}$$

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

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

where X is documents retrieved relevant, Y is documents retrieved irrelevant and Z is documents not retrieved relevant. First F-Measure is calculated locally over each category and then followed by 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 assigns equal weight to each category, irrespective of its frequency.

We have used the SVM light i.e soft-margin linear SVM tool developed by T.Joachims for SVM classification and for KNN classifier, k values range from 5 and taken 10,15,20. In KNN algorithm, we have used the cosine 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 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.

Table 1: Corpus statistics

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

Category	TF	TF-IDF	TF-CHI	TF-RF
Economics	0.712	0.729	0.708	**0.740**
Politics	0.783	0.780	0.759	**0.798**
Science	0.719	**0.740**	0.698	0.731
Sports	0.859	0.867	0.853	**0.875**
Culture	0.814	**0.829**	0.795	0.824
Health	0.875	0.860	0.867	**0.895**
F(macro-averaged)	0.794	0.801	0.780	**0.810**

Table 3: F1 and Macro averaged F1 results of KNN Classifier for six categories

Category	TF	TF-IDF	TF-CHI	TF-RF
Economics	0.719	0.723	0.694	**0.731**
Politics	0.796	0.809	0.799	**0.816**
Science	0.749	**0.761**	0.730	0.753
Sports	**0.902**	0.874	0.869	0.896
Culture	0.851	0.854	0.844	**0.861**
Health	0.885	0.891	0.883	**0.907**
F(macro-averaged)	0.817	0.819	0.803	**0.828**

Table 4: F1 and Macro averaged F1 results of SVM Classifier for six categories

Category	TF	TF-IDF	TF-CHI	TF-RF
Economics	0.726	0.733	0.682	**0.764**
Politics	0.820	0.828	0.812	**0.851**
Science	0.709	**0.751**	0.711	0.747
Sports	0.874	0.890	0.889	**0.915**
Culture	0.848	0.843	0.829	**0.857**
Health	0.917	0.909	0.890	**0.932**
F(macro-averaged)	0.816	0.826	0.802	**0.844**

5. RESULTS AND DISCUSSION

After analyzing the results, we found that the SVM categorizer outperformed NB and KNN on six data sets with regards to F1 and macro averaged-F results. TF-RF performs significantly better for all category distributions. Best macro averaged-F is achieved by using the TF-RF scheme. From the results it is observed that relevance frequency scheme does improve the term's discriminating power for text categorization. It is observation that IDF adds discriminating power TF when combined together. The TF-CHI method has given worse performance than TF,TF-IDF in most of the categories in all classifiers. Moreover, and for the Telugu data sets, the SVM classifier has 1.0%, 1.2% and 2.8% higher macro-averaged F1 than NB, KNN respectively. Another notable result that was also reported is that all classifiers vary among categories. For example, the "Health" category has a neat classification F1 of 93.2%, while the “science” category has a noticeably poor F1 measure of 74.7% for SVM. These poor results indicate that the "Science" category is highly overlapped with other categories.

6. CONCLUSIONS AND FUTURE SCOPE

The macro average F1 of four term weighting measures obtained against six Telugu category sets indicated that the SVM algorithm dominant NB and KNN algorithms. Finally, SVM and KNN classifiers perform excellent in most of the categories.

TF-RF scheme shown good performance compared with other three variants of term frequency. The CHI-square as a factor do not improve the term’s discriminating power for text categorization. With this emperical analysis we are planning to use TF-RF as the term weighing scheme for further research on Telugu Text categorization. Also, planning to propose a hybrid approach, a combination two or more classifiers to increase the accuracy of the text classification process on Telugu documents.

ACKNOWLEDGEMENTS

The authors would like to thank everyone, just everyone!

REFERENCES

[1] YANG Y, PEDERSEN J Q. A comparative study on feature selection in text categorization. Proceedings of the 14th International Conference on Machine Learning (ICML), 1997: 2-3.

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

[3] FABRIZIO SEBASTIANI. Text categorization[M]//Alessandro Zanasi. Text mining and its applications. WIT Press, Southampton, UK, 2005: 110-120.

[4] SALTON G, WONG A, YANG C S. A vector space model for automated indexing. Communications of the ACM, 1975: 1-8.

[5] SALTONG, MCGILLC. An introduction to modern information retrieval. McGraw Hill, 1983.

[6] Yang, Yiming and Pederson, Jan O. 1997. A Comparative Study on Feature Selection in Text Categorization. ICML-97. 412-420.

[7] Tam, Santoso A and Setiono R., “A comparative study of centroid-based, neighborhood-based and statistical approaches for effective document categorization”, ICPR '02 Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) ,vol.4 , no. 4 , 2002, pp.235–238.

[8] Joachims, Thorsten. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. ECML-98. 137-142.

[9] Irina Rish, “An Empirical Study of the Naïve Bayes Classifier”, Proc. of the IJCAI-01 Workshop on Empirical Methods in Artificial Intelligence, Oct 2001. citeulike-article-id:352583.

[10] Doyle Lauren, Joseph Becker, “Information Retrieval and Processing”, Melville, 1975.

[11] Luhn, H.P., "A Statistical Approach to Mechanized Encoding and Searching of Literary Information", IBM J. Res. Develop, 1957.

[12] Manning, Raghavan, Schutze, “Introduction to Information Retrieval”, Cambridge University, 2008 text categorization Feldman, Sanger, “The Text Mining Handbook - Advanced Approaches in Analyzing Unstructured Data”, Cambridge University, 2007.

[13] Robertson, S., “Understanding inverse document frequency: on theoretical arguments for IDF”, Journal of Documentation, vol. 60, pp. 503–520, 2004. machine

[14] Sebastiani F. Machine learning in automated text categorization[J]. ACM Computing Surveys, 2002, 34 (I): 1247.

[15] Sebastiani, F. (1999) ‘A Tutorial on Automated Text Categorisation’, In Amandi , A. and Zunino, A. (eds.), Proceedings of the 1st argentinian Symposium on Artificial Intelligence (ASAI'99), pp. 7-35

[16] Sebastiani, F., Sperduti, A., and Valdambrini, N. 2000. An improved boosting algorithm and its application to automated textcategorization. In Proceedings of CIKM-00, 9th ACM International Conference on Information and Knowledge Management (McLean, US, 2000), pp. 78–85.

[17] Man Lan, Chew Lim Tan, Hwee Boon Low and Sam Yuan Sung. A comprehensive comparative study on term weighting schemes for text categorization with support vector machines. In the Proceedings of 14th International World Wide Web Conference (WWW2005). page 1032–1033. ISBN: 1-59593-051-5. May 2005. Chiba, Japan.