Text Categorization using Term relevance frequency

P.Vijaya pal Reddy
Department of Computer Science and Engineering
Malla Reddy College of Engineering
Hyderabad, India
vijayapalreddy76@gmail.com

Text categorization refers to the process of assign a label among predefined set of labels to each document. Text categorization in Indian languages is challenging as Indian languages are very rich in morphology, giving rise to a very large number of word forms and hence very large feature spaces. This paper investigates the performance of different classification approaches using term relevance frequency as term weight for Telugu text classification problem with NB, SVM and kNN classifiers.

Keywords-Term Relevance measure, Text Categorization, Support Vector Machine, Naive Bayes, k Nearest Neighbor, CHI-square.

I. INTRODUCTION

Text categorization (TC) also known as text classification, is the task of automatically sorting a set of documents into categories (or topics, classes) from a predefined set. Automated text classification tools are attractive since they free the organizations from the need of manual categorization of documents, which can be too expensive.

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]. There are two types of approaches to text categorization: rule based and machine learning based approaches [4]. Rule based approaches mean ones where classification rules are defined manually and documents are classified based on rules. Machine learning approaches mean ones where classification rules or equations are defined automatically using sample labeled documents.

Extensive research works have been not conducted on Telugu corpus since Telugu language is highly rich and requires special treatments such as order 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. To the best of the authors knowledge, there are no comparisons which have been conducted 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 is explained in Section 2. Section 3 describes the the characteristics of Telugu language. Section 4 is dealt with data collection as well as the experimentations. Section 5 is about results analysis, and finally the conclusions and further research are given in Section 6.

II. THE PROPOSED MODEL

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

The proposed system mainly having three modules such that 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 based correct and incorrect classifications for each input. The different phases in the model are explained below:

Tokenization is the process of chopping a document into small units called tokens which usually results in a set of atomic words having a useful 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. A stop list is a list of commonly repeated features which appear in every text document. The

common features such as pronouns, conjunctions and prepositions etc.. need to be removed because they do not have effect on the categorization process. For the same reason, if the feature is a special character or a number then that feature should be 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%.

Stemming is the process of removing affixes (prefixes and suffixes) from features. This process 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. 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. Each input text document is represented as a vector in a vector space, each dimension of this space represents a single feature of that vector and its weight which is computed with different weighting schemes, a point of discussion in this paper, known as vector space model. 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. Term weighting corresponding to a value to a term in order to reflect the importance of that term in a document. There are different term weighting methods proposed in the TC study such as Term Frequency (TF),Term frequency-Inverse Document Frequency (TF-IDF), Term Frequency-Chi square (TF.CHI) and Term Frequency-Relevance Frequency (TF.RF). The relevance measure is calculated as follows:

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

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

In Indian languages, the number of features will be even higher compared with English text because of richness in morphology. We use $\chi 2$ metric [1] for feature selection in this paper, which are found $\chi 2$ and information gain are the most effective feature selection metrics in the literature. CHI square measures the correlation between feature and class. 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 statistics can be depicted as:

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

III. EXPERIMENTAL EVALUATIONS

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 70% of the documents as training samples, remaining 30% of the documents as testing samples for all six categories. Then we use CHI square statistics feature selection method to select 100 features. The experiments were conducted using relevance frequency as a term weighing measure on various classifiers such as Naive Bayes, KNN and SVM.

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:

$$F_1 = \frac{2 * Recall * Precision}{Recall + Precision}$$ 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. 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 SVM light 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 for F1 values results of various classifiers for six categories.

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

Category	K-Nearest Neighbor	Support Vector Machine	Naive Bayes
Economics	0.731	0.764	0.740
Politics	0.816	0.851	0.798
Science	0.753	0.747	0.731
Sports	0.896	0.915	0.875
Culture	0.861	0.857	0.824
Health	0.907	0.932	0.895
F(macro-averaged)	0.828	0.844	0.810

Table 3: F1 and macro averaged F1 value results of NB, KNN and SVM classifiers for six categories

IV. RESULTS ANALYSIS

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. Moreover, and for the Telugu data sets, the SVM classifier have 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 "Sport" category has a neat classification F1 of 91.5%, 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.

V. 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.

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 1 0 3 2 – 1 0 3 3 . I S B N : 1 - 5 9 5 9 3 - 0 5 1 - 5 .