ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

A Neoteric Method for the Classification of Text Using SVM Classifier

Mohammed Asrar Naveed , Chaitra B, Priya Dahiya

M. Tech Student, Dept. of ISE, Acharya Institute of Technology, Bengaluru, India

Assistant Professor, Dept. of ISE, Acharya Institute of Technology, Bengaluru, India

M. Tech Student, Dept. of ISE, Acharya Institute of Technology, Bengaluru, India

ABSTRACT: in today's world, browsing for exact information has become very tedious job as the number of electronic documents on the Internet has grown big and large. It is necessary to classify the documents into categories so that retrieval of documents becomes easy and more efficient. We try to overcome this difficulty by efficiently organizing, the documents into set of related topics or categories, which enable the user query process, to be precise and optimized. Main goal of this paper is to classify documents into a certain number of predefined categories. We have used k-means clustering algorithm, tf-idf feature extraction for indexing and SVM classifier.

KEYWORDS: K-means clustering, TF-IDF, SVM classifier

I.**INTRODUCTION**

Basically document classification can be defined as content based assignment of one or more predefined categories or topics to documents i.e., collection of words determine the best fit category for this collection of words. The goal of all document classifiers is to assign documents into one or more content categories such as technology, entertainment, sports, politics, etc., Classification of any type of text document is possible, including traditional documents such as memos and reports as well as e-mails, web pages, etc. Text Classification is also called Text Categorization or Document Classification. Document classifications are of two approaches manual classification and automatic classification.

Text classification effort involuntarily to decide whether a document or part of a document has exacting distinctiveness typically predestined on whether the document enclose a convinced type of topic or not. Uniquely, the topic of fascination is not defined absolutely or more accurately by the users and as a substitute they are arranged with a set of documents that comprise the enthrallment of both (positive and negative training set). The decision-making procedure despotically hauls out the features of text documents and helps to discriminate positive from negatives. It also administers those appearances inevitably to sample documents with the help of text classification systems. The k-nearest neighbor rule is a simple and effective classifier for document classification. In this technique, a document is fashionable into an exacting category if the category has utmost diversity pertaining to the k nearest neighbors of the documents in the training set. The k nearest neighbors of a test document is ordered based on their content similarity with the documents in the training set.

Automatic text classification has several useful applications such as classifying text documents in electronic format; spam filtering; improving search results of search engines; opinion detection and opinion mining from online reviews of products, movies or political situations and text sentiment mining. Blogging has become a popular means of communication over the Internet. New abbreviations, slang terms etc are added on a daily basis on blogs, which are in turn quickly accepted by the blog users. In order to implement text classification applications like opinion mining or sentiment classification, it is required to keep track of such newly emerging terms (not found in standard language dictionaries). The nature of blog entries is such that additional content is added on a daily basis. Moreover, text posts on a blog do not strictly adhere to the blog topic. This introduces the need to develop incremental and multi-topic text classification techniques. There is also the need to develop automated, sophisticated text classification and summarization tools for many regional languages as several blogs and newspaper sites in these languages have become popular.

II. Related Work

Xiaoyan Cai et .al [1] have discussed reinforcement after relevance propagation (RARP) i.e. manifold ranking based relevance propagation with mutual reinforcement between sentences and clusters. They ranked a sentence higher if it is contained in the theme cluster which is more relevant to the given query while a theme cluster ranked higher if it contains many sentences which are more relevant to the given query. Different from the traditional query-focused summarization approaches, which were either the simple extensions of generic summarizers and did not uniformly fuse the information in the query and the documents, or based on semi-supervised learning methods and/or supervised learning methods.

Kogilavani et al. [6] have considered the feature profile for sentences. Feature profile is generated by considering word weight, sentence position, sentence length, and sentence centrality, proper noun in the sentence and numerical data in sentence.

Zha [9] proposed a mutual reinforcement principle that was capable of extracting significant sentences and key phrases at the same time. In his work, a weighted bipartite document graph was constructed by linking together the sentences in a document and the terms appearing in those sentences. The mutual reinforcement was reduced to a solution for the singular vectors of the transition matrix of the bipartite graph. The relevance of each text unit to the given query was calculated by the cosine similarity and characterized by the corresponding text vertex in a three-layer text graph.

Basu et.al [10] assigned a document to a predefined class for a large value of k when the margin of majority voting is one or when a tie occurs. The majority voting method discriminates the criterion to prune the actual search space of the test document. This rule has enhanced the confidence of the voting process and it makes no prior assumption about the number of nearest neighbors.

III. Proposed System

Figure1 shows the block diagram of our proposed system. It includes training and testing phase. In training phase, few text files are considered and include pre-processing, feature extraction, indexing and SVM classifier. In testing phase text classification is made to the user given text file as the input.

Input Data: Text files are taken as the input to our proposed system.

Pre-Processing: The system takes all types of text documents i.e. .txt, .pdf, .rtf, .doc, .html etc. and query as input. Firstly it converts all documents in .txt files. Then it tokenizes the text documents in order to find the individual terms. Then filtering of the text is done by removing the stop words and remaining words are stemmed using Porter Stemmer algorithm. The term weight is calculated as follows, Term Weight = tf * idf where, tf – term frequency idf - inverse document frequency, after this the documents are grouped according to entered query. After grouping the documents, the next job is of scoring of sentences based on feature profile.

Tokenization: Tokenization [2] is the method of splitting a stream of text input into meaningful elements. These meaningful elements are called tokens like symbols, phrases, words so on. The extracted group of tokens acts as an input for further processes like parsing and text mining. Usually, the tokenization process occurs at the word level. Yet to define what is meant by a "word" is sometimes difficult to deal with, often tokenizes relies on some simple heuristics. This can be made clear with some examples such as:

- All adjacent strings of alphabetic characters are always a part of one token. The same is the case numbers.
- Tokens may be separated by whitespace characters. These may include punctuation characters, a line break or space.
- The resulting list of tokens may or may not include Punctuation and whitespace.

Stopwords [5]: The next process in this step is to reduce the size of the list created by the parsing process, generally using methods of stop words removal and stemming. Stop words are removed from each of the document by comparing the one with the stop word list. This process reduces the number of words in the document significantly since these stop words are insignificant for search keywords. Stop words can be pre-specified list of words or they can depend on the context of the corpus.

Stemming: The next process in phase one after stop word removal is stemming[4]. Stemming is a process of linguistic

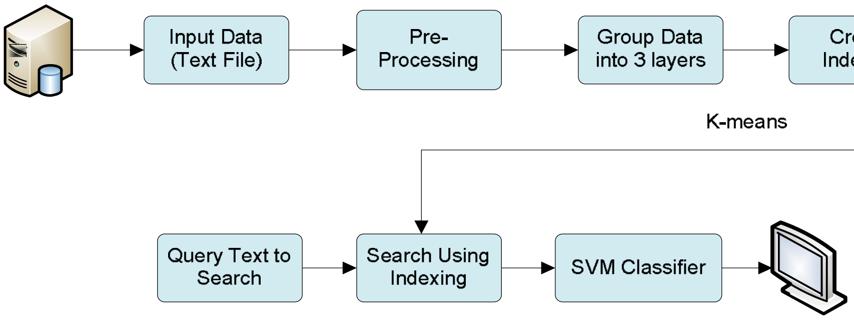

Figure 1: Shows the Block Diagram of proposed system.

Normalization in which the variant forms of a word is reduced to a common form.

a) *Grouping the data into Layers:*

Here we are using K-means algorithm to partition group of data into 3 layers or clusters. Initially Partitioning methods[8] work by dividing a group of n elements into 3 clusters such that 3 is less than or equal to n and each cluster contains at least one element. Partitioning methods conduct one level partitioning on data sets. It iteratively improves the clusters by relocating objects from one group to a more relevant one. It continues this process until the clusters stabilize and no more migration of data from one cluster to another takes place. Partitioning method used is k-means

K-Means Algorithm

In this algorithm the data is divided into a pre-specified k clusters. The result obtained is such that there is at least one item in each cluster all of which does not overlap and are non-hierarchical. The algorithm works as follows:

1. First divide the given data set into k partitions assigning each partition roughly the same number of elements.
2. Next, find the mean of the elements of each partition.
3. Check whether each element in each partition is closer distance wise to the mean of the current cluster or to the mean of another cluster. In an event where the element is closer to the mean of another cluster, relocate the element to that cluster. Perform this check for every element in each partition.
4. Repeat process 3 until a set of stable clusters has been obtained and further relocation of elements is not possible. This method, however, is sensitive to outliers as it takes into consideration, the mean of each cluster.

b) *Creating Indexing Using Feature extraction*

A feature selection[7] method only selects a subset of meaningful or useful dimensions specific to applications from the original set of dimensions. The text mining domain, attribute feature selection methods include Document Frequency (DF), Term Frequency (TF), Inverse Document Frequency (IDF) etc. These methods fort requisites based on numerical procedures computed for each document in the text corpus. The presence of terms in documents needs to be recorded in the index, a term can also be assigned a weight that expresses its importance for a particular document. A commonly used term weighting method is tf-idf, which assigns a high weight to a term, if it occurs frequently in the document but rarely in the whole document collection. Contrarily, a term that occurs in nearly all documents has hardly any discriminative power and is given a low weight, which is usually true for stopwords like determiners, but also for collection-specific terms; thinks of apparatus, method or claim in a corpus of patent documents. For calculating the tf-idf weight of a term in a particular document, it is necessary to know two things: how often does it occur in the document (term frequency = tf), and in how many documents of the collection does it appear (document frequency = df) and the total number of documents = N. Take the inverse of the document frequency (= idf) and you have both components to calculate the weight by multiplying tf by idf.

$$idf = log\frac{N}{df}$$

After performing the feature extraction process, the extracted features are used to classify the input text data.

DOI: 10.15680/IJIRCCE.2016. 0406237

After feature selection, the text document is represented as a document vector, and an appropriate machine learning algorithm is used to train the text classifier. The trained classifier is tested using a test set of text documents. If the classification accuracy of the trained classifier is found to be acceptable for the test set, then this model is used to classify new instances of text documents.

c) SVM Classifier

Support Vector Machine is an advanced supervised modeling technique for classifying both linear and nonlinear data[3]. SVM has become a more recent default approach to classification problems since it is well suited to very high-dimensional spaces and extremely big datasets. In machine learning, support vector machines are supervised learning models with associated learning algorithms. They perform actions such as analyze data and recognize patterns. They are used for regression analysis and classification. The simple SVM takes a set of input data. Then for each given input, it predicts, which of two possible classes forms the output which makes it a non-probabilistic binary linear classifier. First the support vector machine is trained using the training value sets prepared according to our relevant domain and their texts. After training, the trained machine is used to classify or predict a new input text data.

In testing phase, when a User inputs the query text to search three main steps are performed. Firstly data pre-processing is performed which involves tokenizing and stemming of the input text. Secondly, Feature extraction is performed. Thirdly, after extracting the features classification is done. SVM classifier can be chosen to identify the relevant domain names for the query is shown.

IV. RESULTS AND DISCUSSION

a) *Data reduction:* is the process of minimizing the amount of data. In our proposed system we are using k-means Clustering technique to reduce the amount of data for classification purpose. Data reduction can increase storage efficiency and reduce costs. Figure 2 shows the comparison graph for reduction rate vs. training load.
b) *Execution time* is the time during which a program is running. How fast the task is completed by our proposed system i.e., time taken for classification of text file. Figure 3 shows the graph of execution time taken for our proposed system to existing system.

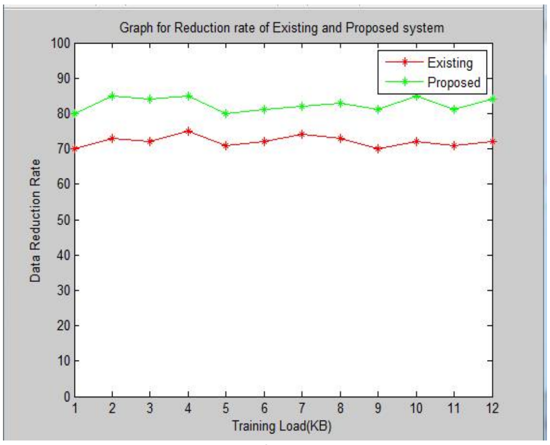

Figure 2: Graph for Reduction Rate

DOI: 10.15680/IJIRCCE.2016. 0406237

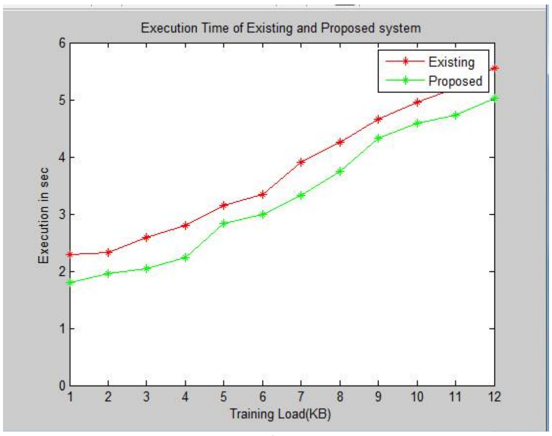

Figure 3: Graph for Execution Time.

V. CONCLUSION

An Automated text classification system is attractive because it frees organizations from the need of manually organizing documents that can be too expensive, which otherwise may not be feasible with the given time constraints of the application when more number of documents are involved. In this paper we have proposed a text classification system using support vector machine, k-means cluster and tf-idf feature. Our results gave good execution time when compared to existing methods and also less data reduction rate.

REFERENCES

[1] XiaoyanCai and Wenjie Li "Mutually Reinforced Manifold-Ranking Based Relevance propagation Model for Query-Focused Multi-Document Summarization" IEEE Transactions on audio, speech, and language processing, Vol. 20, No. 5, 2012.

[2] Rene Arnulfo Garcia-Herandez and Yulia Ledeneva,"Word Sequence Models for Single Text Summarization",IEEE,pp.44-48, 2009.

[3] Yongzheng, Nur and Evangelos,"Narrative Text Classification for Automatic Key Phrase Extraction in Web Document Corpora", WIDM'5, Bremen Germany, pp.51-57, 2005.

[4] Dragomir R. Radev, Hongyan Jing and Molgorzata Stys,"Centroid-based summarization of multiple documents", International Journal of Information Processing and Management, 2004.

[5] A. P. Siva Kumar, Dr. P. Premchand and Dr. A. Govardhan "Query–based summarizer based on similarity of sentences and word frequency", International journal of Data Mining & Knowledge Management Process (IJDKP) Vol. 1,No.3,2011.

[6] A. Kogilavani and Dr. P.Balasubramani, "Clustering and feature specific sentence extraction based summarization of multiple documents", International journal of computer science and information technology (IJCSIT) Vol2. No.4 , 2010.

[7] X. J.Wang, J. W. Yang, and J. G. Xiao, "Manifold-ranking based topic focused multidocument summarization," in Proc. 18th IJCAI Conf., pp. 2903–2908, 2007.

[8] Rodolfo P. dos Santos, Gabriela S. Clemente, Tsang Ing Ren and George D.C. Calvalcanti, "Text Line Segmentation Based on Morphology and Histogram Projection", 2009.

[9] H. Y. Zha, "Generic summarization and key phrase extraction using mutual reinforcement principle and sentence clustering," in Proc. 25th SIGIR Conf., pp. 113–120,2002.

[10] M.C. Padma and P.A.Vijaya, "Entropy Based Texture Features Useful for Automatic Script Identification", Volume 2, Issue 2, pp 115-120, 2010.