International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 02 | Feb -2017 www.irjet.net p-ISSN: 2395-0072

Text Document categorization using support vector machine

Shugufta Fatima, Dr. B. Srinivasu

Shugufta Fatima M.Tech

Dept. of Computer Science and Engineering , Stanley College of Engineering and Technology for Women, Telangana- Hyderabad, India.

Dr. B. Srinivasu Associate Professor

Dept. of Computer Science and Engineering , Stanley College of Engineering and Technology for Women, Telangana- Hyderabad, India.

Abstract - *The Web is a tremendous source of information, so tremendous that it becomes difficult for human beings to select meaningful information without support. Categorization of documents refers to the problem of automatic classification of a set of documents in classes (or categories or topics). Automatic Text Categorization is an important issue in the text mining. The task is to automatically classify text documents into predefined classes based on their content.*

Automatic categorization of text documents has become an important research issue now a days. Proper categorization of text documents requires information retrieval, machine learning and Natural language processing (NLP) techniques. Our aim is to focus on important approaches to automatic text categorization based on machine learning technique.

Several methods have been proposed for the text documents categorization. We will adapt and create machine learning algorithms for use with the Web's distinctive structures: large-scale, noisy, varied data with potentially rich, human-oriented features using svm.

Key Words: Text Documents Categorization, Machine Learning, Support Vector Machine.

1. INTRODUCTION

1.1 Document categorization

The Web is a tremendous source of information, so tremendous that it becomes difficult for human beings to select meaningful information without support. Automatic Text Categorization is an important issue in the text mining. The task is to automatically classify text documents into predefined classes based on their content [1]. Feature selection and learning are two important steps in the automatic text categorization. Classification algorithms take a "training" set of labeled documents and tries to find a relationship between the labels and a set of features, such as the words in the documents.

1.2 Text Categorization

Text Categorization is the process of assigning a given text to one or more categories. This process is considered as a supervised classification technique, since a set of pre-classified documents is provided as a training set. The goal of Text Categorization is to assign a category to a new document.

1.3 Problem Description

Information is mostly in the form of unstructured data .as the data on the web has been growing, it has lead to several problems such as increased difficulty of finding relevant information and extracting potentially useful knowledge. As a consequence of this exponential growth, great importance has been put on the classification of documents into groups that describe the content of the documents. The function of a classifier is to merge text documents into one or more predefined categories based on their content.
Unlabeled texts provide co-occurrence information for words, which can be used to improve categorization performance. Although unlabeled texts are available from the internet, collecting unlabeled texts which are useful for a text categorization problem is not an easy task because of the wide diversity of texts on the Internet.

1.4 Basics and background knowledge

1.4.1 Approaches for Classification

Classification is a problem where a learner is given several labeled training examples and is then asked to label several formerly unseen test examples.this can be done in three ways

- Supervised text classification
- Unsupervised text classification
- Semi-supervised text classification

1.4.1.1 Supervised Text Classification

One of the classification approach is supervised text classification it differs from unsupervised in the way the classes are divided, it takes labeled documents and additional knowledge is provided by an expert.

These algorithms use the training data, where each document is labeled by zero or more categories, to learn a classifier which classifies new texts. A document is considered as a positive example for all categories with which it is labeled, and as a negative example to all others. The task of a training algorithm for a text classifier is to find a weight vector which best classifies new text documents .The different approaches for supervised text classification can be summarized as

A. K-Nearest Neighbor classifier

B. Naïve Bayes Method

C. Decision Trees

D. Decision Rules Classification

E. Support Vector Machines

1.4.1.2 Unsupervised Text Classification

Another approach for classification is unsupervised classification and it takes unlabelled documents and additional knowledge is not provided by expert. The aim is to cluster the documents without additional knowledge or intervention such that documents within a cluster are more similar than documents between clusters. These are categorized into two major groups as partitioned and hierarchical.

A. Hierarchical Clustering Techniques
1.Divisive Hierarchical Clustering

2.Agglomerative Hierarchical Clustering

- Single-link
- Complete-link
- Average-link

Hierarchical algorithms produce nested partitions of data by splitting (divisive approach) or merging (agglomerative approach) clusters based on the similarity among them.

B. Partition Clustering Techniques

Partitioned clustering algorithms group the data into un-nested non-overlapping partitions that usually locally optimize a clustering criterion.the algorithms under this are

- *K-Means Clustering*
- *Bisecting K-Means*

C. Kohonen's Self Organizing Network

1.4.1.3 Semi-supervised Text Classification

These algorithms make use of unlabeled data along with few labeled data to classify new unlabeled text document. In text classification most of the times there is limited labeled data, and in most cases it can be expensive to generate that labeled data so semi-supervised algorithms gives good solution in such a situations. Its framework is applicable to both classification and clustering. Some of the important algorithms discussed here are as

A. Co-training

B. Expectation Maximization

C. Graph based

2. LITERATURE SURVEY

2.1 Introduction

Documents come from a wide variety of sources. Many are generated with various word processing software packages, and are subjected to various kinds of automatic scrutiny, e.g., spelling checkers, as well as to manual editing and revision. Many other documents, however, do not have the benefit of this kind of scrutiny, and thus may contain significant numbers of errors of various kinds. Email messages and bulletin board postings, for example, are often composed on the fly and sent without even the most cursory levels of inspection and correction.

World Wide Web(WWW) is widely used to access information on the internet. Many web sites are automatically populated by using common templates with contents. For human beings, the templates make readers easy to access the contents guided by consistent structures, even though the templates are not explicitly announced. However, for machines, the unknown templates are considered harmful because they degrade the accuracy and performance due to the irrelevant terms in templates. Thus, template detection and extraction techniques have received a lot of attention recently to improve the performance of web applications such as data integration, search engines, classification of web documents and so on.

2.2 Related works

Many types of text classification have been proposed in the past. A well known one is the bag of words that uses keywords (terms) as elements in the vector of the feature space. In order to reduce the number of features, many dimensionality reduction approaches have been conducted by the use of feature selection techniques, such as Information Gain, Mutual Information, Chi-Square, Odds ratio, and so on. Details of these selection functions were stated in [2,3].

In [4], Nigam K. demonstrated that supervised learning algorithms that use a small number of classified documents and many inexpensive unclassified documents can create high accuracy text classifiers.

In [5], Ifrim G. proposed a model to text categorization that concentrates on the underlying meaning of words in their context (i.e., concentrates on learning the meaning of words, identifying and distinguishing between different contexts of word usage). This model can be summarized in the following steps:

- Map each word in a text document to explicit concepts.
- Learn classification rules using the newly acquired information.
- Interleave the two steps using a latent variable model.

3. ARCHITECTURE

3.1 System architecture:

In classification the document class labels are an essential part of the input to the learning system. The objective is to create a model between a set of documents and a set of class labels. This model is used to determine automatically the class of new documents. This mapping process is called classification. The general framework for classification includes the model creation phase and other steps. Therefore the general framework is usually called supervised learning (learning from examples) This includes following steps.

1. Documents are collected cleaned and properly organized, the terms(features) identified, and a vector space representation created. Data divided into two subsets: training set: this part of the data will be used to create the mode. Test set: this part of the data is used for testing the model.
2. Building the model:
   This is the actual learning(also called training) step, which includes the use of the learning algorithm. it is usually an iterative and interactive process that may include other steps and may be repeated several times so that the best model is created; feature selection, applying the learning algorithm, validating the model(using the validation subset to tune some parameters of the learning algorithm)
3. Testing and evaluating the model:
   At this step the model is applied to the documents from the test set and their actual class labels are compared to the labels predicted.
4. Using the Model to Classify New Documents(with unknown class labels).

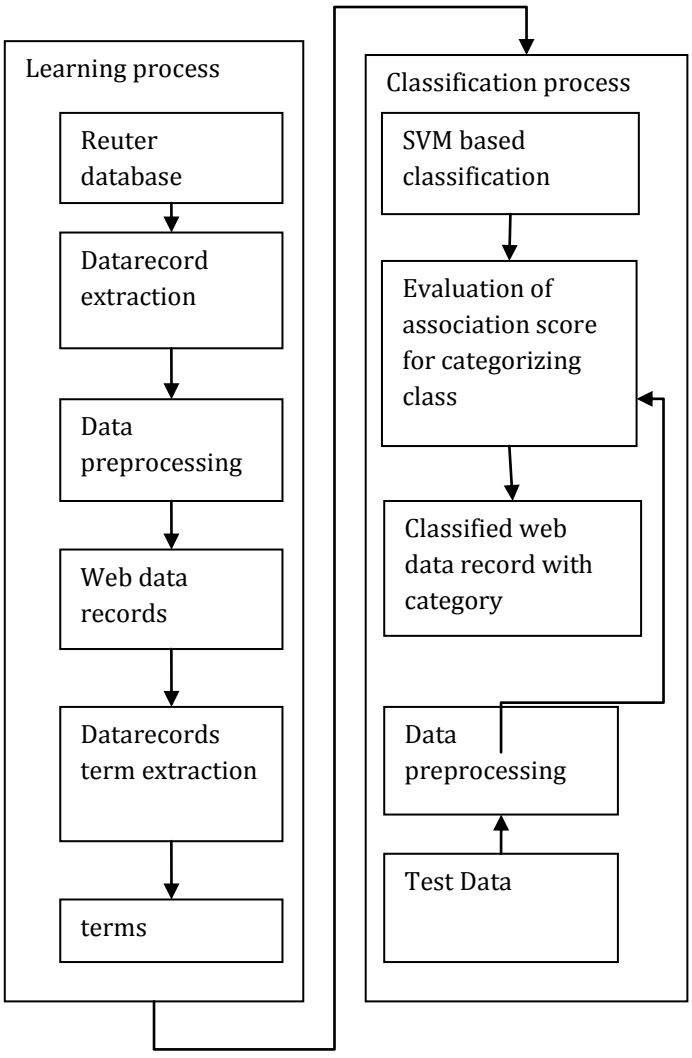

Figure 3.1 Architecture

3.2 System modules

Based on the proposed system architecture in Figure-3.1, we construct the methodology into three modules as follows:

1. Learning Processing
2. Term Extraction
3. Classification

3.2.1 Learning process:

This module processes the texts of the text database into standard format and extracts multi-word features. It implements three preprocessing methods as follows, and it's the training phase.

Stop word elimination.

Stop words, or stop words, is a name given to words which are filtered out prior to the processing of natural language data (text). They are generally regarded as 'functional words' which do not carry meaning.

Stemming word elimination using Porter Algorithm

This algorithm reduces all words with the same root to a single form, the stem, by stripping the root of its derivational and inflectional affixes; in most cases, only suffixes that have been added to the right-hand end of the root are removed and this approach to conflation forms.

3.2.2 Term Extraction:

This module performs the extracting of the multi-term from the text document we implements *tf-idf* method. Multi-term is neither derived from a classical mathematic model nor a formally defined linguistic model. Although it is superiority in text classification could be explained using term frequency.

We produce multi-term candidates by matching any two sentences in a document to look for the repetitive patterns, After repetition extraction from documents, we should normalize these repetitions to meet the regular expression. The procedure implements used to normalize the repetitions to get the multi-terms extraction.

3.2.3 classification :

This module implements the Classification process to perform the classification based on the generated knowledge or training data using the SVM Classification Approach, the generated output for the web data record categorization.

Text categorization is the task of deciding whether a piece of text belongs to any of a set of pre -specified categories. It is a generic text processing task useful in indexing documents for later retrieval, as a stage in natural language processing systems, for content analysis, and in many other roles. The use of standard, widely distributed test collections has been a considerable aid in the development of algorithms for the related task of text retrieval (finding documents that satisfy a particular user's information need, usually expressed in a textual request). Text retrieval test collections have allowed the comparison of algorithms developed by a variety of researchers around the world. We use a standard distributed dataset for our evaluation from Reuters-21578 collection datasets. We will discuss on this implementation section.

4. IMPLEMENTATION

4.1 Learning process

PreProcessing :

- **Tokenization**

The process of breaking a stream of text into words is called tokenization. In Java the StringTokenizer class allows an application to break a string into tokens.

- **Eliminating Stop Words**

Stop words, or stop words, is a name given to words which are filtered out prior to the processing of natural language data (text). They are generally regarded as 'functional words' which do not carry meaning. By our observation concerning the occurring positions of the stop words in a sentence, three kinds of full matching of stop words are programmed to eliminate the stop words as "stop word + white space" for the beginning position, "white space + stop word + punctuation" for the end position and "white space + stop word + white space" for the middle position while "+" means "followed by".

- **Eliminating Stemming Words**

Removing suffixes by automatic means is an operation which is especially useful in the field of information retrieval. In a typical IR environment, one has a collection of documents, each described by the words in the document title and possibly by words in the document abs.here we use porters algorithm.

4.2 Term Extraction

Text data is composed of terms grouped into sentences and paragraphs. One technique to represent text data using the vector space model breaks down the textual data into a set of terms. Each of these terms corresponds to an attribute of the input data and therefore becomes an axis in the vector space. The data is then represented as vectors, whose components correspond to the terms contained in the data collection and whose value indicates either a binary present/absent value or a weightage for the term for the data point. This representation is known as the Extracted Term' representation and is the most commonly used representation for text data patterns. Pattern mining has been extensively studied in data mining communities for many years.

- **Term Weighting:**

In the vector space model, the documents are represented as vectors. Term weighting is an important concept which

determines the success or failure of the classification system. Since different terms have different level of importance in a text, the term weight is associated with every term as an important indicator . Term frequency of each word in a document (TF) is a weight which depends on the distribution of each word in documents. It expresses the Importance of the word in the document. Inverse document frequency of each word in the document database (IDF) is a weight which depends on the distribution of each word in the document database. It expresses the importance of each word in the document database . TF/IDF is a technique which uses both TF and IDF to determine the weight a term. TF/IDF scheme is very popular in text classification field and almost all the other weighting schemes are variants of this scheme. In vector space model organization of document also affect the performance of system . In this experiment we use term frequency method, other are also acceptable.

TF-IDF, weights a given term to determine how well the term describes an individual document within a corpus. It does this by weighting the term positively for the number of times the term occurs within the specific document, while also weighting the term negatively relative to the number of documents which contain the term. Consider term t and document d 2 D, where t appears in n of N documents in D. The TF-IDF function is of the form:

TFIDF(t, d, n,N) = TF(t, d) × IDF(n,N)…….. (4.2.1)

There are many possible TF and IDF functions. Practically, nearly any function could be used for the TF and IDF. Regularly-used functions include:

$$TF(t,d) = \begin{cases} 1 & \text{if } t \in d \\ 0 & \text{else} \end{cases} \quad \text{.....(4.2.2)}$$

$$TF(t,d) = \sum_{word \in d} \begin{cases} 1 & \text{if } word = t \\ 0 & \text{else} \end{cases} \quad \text{..... (4.2.3)}$$

Additionally, the term frequency may be normalized to some range. This is then combined with the IDF function. Examples of possible IDF functions include:

$$IDF(n,N) = \log(N|n) \quad \text{.....(4.2.4)}$$

$$IDF(n,N) = \log(N - n|n) \quad \text{.....(4.2.5)}$$

Thus, a possible resulting TFIDF function could be:

$$TFIDF(t, d, n,N) = \sum_{word \in d} \begin{cases} 1 & \text{if } word = t \\ 0 & \text{else} \end{cases} \text{ X } \log(N - n|n) \text{.....(4.2.6)}$$

When the TF-IDF function is run against all terms in all documents in the document corpus, the words can be ranked by their scores. A higher TF-IDF score indicates that a word is both important to the document, as well as relatively uncommon across the document corpus. This is often interpreted to mean that the word is significant to the document, and could be used to accurately summarize the document . TF-IDF provides a good heuristic for determining likely candidate keywords, and it (as well as various modifications of it) have been shown to be effective after several decades of research. Several different methods of keyword extraction have been developed since TF-IDF was first published in 1972, and many of these newer methods still rely on some of the same theoretic backing as TF-IDF. Due to its effectiveness and simplicity, it remains in common use today .

4.3 Classification
SVM Based Classification:

SVM is a Dedcriminative classifier formally defined by a separating hyperplane.in other words, given labelled trainging data(supervised learning),the algorithm outputs categories.

Support Vector Machines (SVM) is a relatively new class of machine learning techniques first introduced by Vapnik and has been introduced in TC by Joachims. Based on the *structural risk minimization* principle from the computational learning theory, SVM seeks a decision surface to separate the training data points into two classes and makes decisions based on the *support vectors* that are selected as the only effective elements in the training set.

Given a set of *N* linearly separable points $S = \{xi \varepsilon Rn | i = 1, 2, \ldots, N\}$, each point *xi* belongs to one of the two classes, labeled as $yi \in \{-1, +1\}$. A *separating hyper-plane* divides *S* into 2 sides, each side containing points with the same class label only. The *separating hyper-plane* can be identified by the pair (*w,b*) that satisfies

$$w \,.\, x + b = 0$$

and

$$\begin{cases} w \cdot x_i + b \geq +1 \text{ if } y_i = +1 \\ w \cdot x_i + b \leq -1 \text{ if } y_i = -1 \end{cases}$$

for $i = 1, 2, \ldots, N$; where the dot product operation(.) is defined by

$$w \,.\, x = \sum_i w_i x_i$$

for vectors *w* and *x*. Thus the goal of the SVM learning is to find the *optimal separating hyper-plane* (*OSH*) that has the maximal margin to both sides. This can be formularized as:
minimize ½ w.w subject to

$$\begin{cases} w \cdot x_i + b \geq +1 \text{ if } y_i = +1 \\ w \cdot x_i + b \leq -1 \text{ if } y_i = -1 \end{cases} \quad \text{for } i = 1, 2, \ldots, N$$

Figure 4.3.1 shows the *optimal separating hyper-plane*.

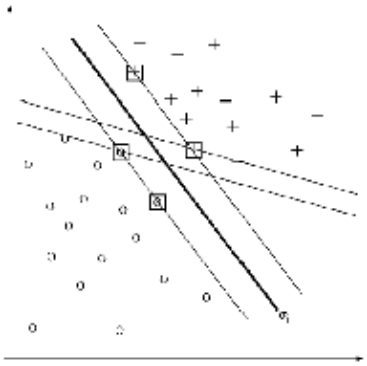

Figure 4.3.1 Learning support vector classifiers.

The small crosses and circles represent positive and negative training examples, respectively, whereas lines represent decision surfaces. Decision surface σi (indicated by the thicker line) is, among those shown, the best possible one, as it is the middle element of the widest set of parallel decision surfaces (i.e., its minimum distance to any training example is maximum). Small boxes indicate the support vectors.
During classification, SVM makes decision based on the *OSH* instead of the whole training set. It simply finds out on which side of the *OSH* the test pattern is located. This property makes SVM highly competitive, compared with other traditional pattern recognition methods, in terms of computational efficiency and predictive accuracy.

5. EXPERIMENT RESULTS

Table 5.1 documents considered for training and testing purpose are illustrated in the below

Documents considered	No. of documents
Training purpose	800
Testing purpose	200

Table 5.2 Evaluation of sample documents from reuters dataset

Reuter	threshold	records	classified	Non-classified	accuracy
reut2001.sgm	50	169	146	23	86.39
reut2001.sgm	60	169	142	27	84.02
reut2001.sgm	70	169	140	29	82.84
reut2001.sgm	80	169	136	33	80.47
reut2001.sgm	90	169	118	51	69.82
reut2001.sgm	100	169	112	57	66.27

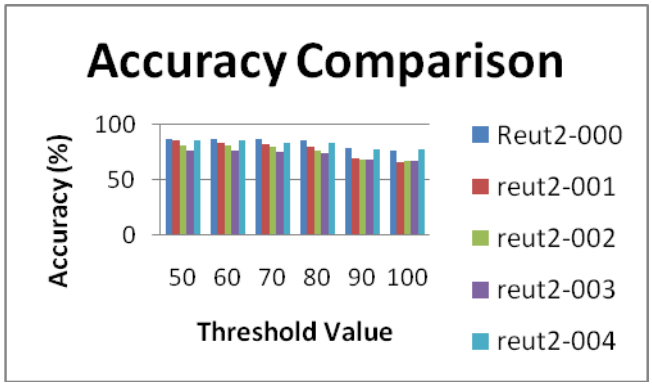

Figure 5.3 Comparision of Accuracy with respect to Threshold Value.

6. CONCLUSIONS

This report presented the results of classifying reuter corpora by using a support vector machine technique. It's a supervised learning technique in machine learning approach.

Support vector machines are binary classifiers, it's a new method for classification of both linear and non-linear data ,it works well with high dimensional data.

In this proposed report we are taking the reuter datasets documents and prepossessing the data by performing string tokenizer, stopword removal, stemmer(porter algorithm) and then term extraction is performed using term frequency technique and the resultant are considered as features using vector space model, the resultant data which we have trained is passed to svm which classifies the data and scoring is performed and the terms are classified into pre defined categories. the test data is pre processed and passed and tested and accuracy check is performed on result.

As we are partitioning documents into training and testing sets with an ratio of 60-40 – 80-20 the accuracy increases, in general it gave better accuracy for 80 training documents and 20 testing documents division.

In our future work, we can introduce other classification algorithms in addition to the ones used here. We can expand it further to utilize other weighting methods and classification techniques and compare them.

ACKNOWLEDGEMENT

To Mummy

Their's nothing that i can ever repay for what you have done to me. Thank you mummy for your unconditional blessings, love, care, support and your teachings in making my life meaningful.

To Pappa

I see your prayers in every wonderful thing that happens to me. you have always been my biggest encouragement and dream supporter. You cared for my education and life and i know and believe that your plans are blessings in my life. Thank you papa for your unconditional support ,care, love , effection.

It is with a sense of gratitude and appreciation that I feel to acknowledge any well wishers for their king support and encouragement during the completion of the project.

I would like to express my heartfelt gratitude to my Project guide **Dr. Srinivasu Badugu,** Coordinator of M.Tech in the Computer Science Engineering department, for encouraging and guiding me through the project. His extreme energy, creativity and excellent domain knowledge have always been a constant source of motivation and orientation for me to gain more hands-on experience and hence get edge over. I am highly indebted to him for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project.

REFERENCES

1. F. Camastra, A. Ciaramella, A. Placitelli, and A. Staiano, "Machine Learning-based Web Documents Categorization by Semantic Graphs", ResearcGate Publisher, DOI: 10.1007/978-3-319-18164-6_8, June-2015.

2. Nigam, K. (May 2001). Using Unlabeled Data to Improve Text Classification. PhD Thesis, School of Computer Science, Carnegie Mellon University, USA.

3. Hayes, P. and Weinstein, S. (1990). Construe/tis: a system for content-based indexing of a database of news stories. In Annual Conferenceon Innovative Applications of AI.

4. Kamal Nigam, Andrew Mccallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using em. Machine Learning, 39(2/3):103–134, 2000.

5. Ifrim, G. (February 2005). A Bayesian Learning Approach to Concept-Based Document Classification. M.Sc Thesis, Computer Science Dept., Saarland University, Saarbrücken, Germany.

6. Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In Proceedings of the European Conference on Machine Learning, pages 137–142, 1998

7. Fabrizio Sebastiani , Machine Learning in Automated Text Categorization, ACM Computing Surveys, Vol. 34, No. 1, March 2002, pp. 1–47.

8. Sebastiani F. 2002. Machine Learning in Automated Text Categorization, ACM Computing Surveys, 34(1): 1-47.

9. Korde, V.& Mahender, C. (March 2012). Text Classification and Classifiers: A Survey. International Journal of Artificial Intelligence & Applications (IJAIA), Vol. 3, No. 2, pp. 85-99.

10. Lee, K. (September 2003). Text Categorization with a Small Number of Labeled Training Examples. PhD Thesis, School of Information Technologies, University of Sydney, Australia.

11. Hirotoshi Taira and Masahiko Haruno. Feature selection in SVM text categorization. In Proceedings of the 16th National Conference on Artificial Intelligence / Eleventh Conference on Innovative Applications of Artificial Intelligence (AAAI99/IAAI-99), pages 480–486, 1999.

BIOGRAPHIES

Shugufta fatima received her B.E and M.tech degree in computer science from Osmania university. Her research interest are machine learning and semantic web.

Dr.B.Srinivasu received his Ph.D degree from University of Hyderabad and B.Tech, M.Tech degree in computer science from JNT University. His research interest are Natural language processing,text processing,Big data.