www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 5 may, 2014 Page No. 6022-6026

Realization of computerized text taxonomy through a supervised learning system

Mr. Suresh G S, Mrs. Sharayu Pradeep
(Master of Technology)
Department of Computer Science and Engineering, Channabasaveswara Institute Of Technology,
Tumkur – 572216, Karnataka, India
(gssuresh@cittumkur.org)
Assistant Professor,
Department of Computer Science and Engineering, Channabasaveswara Institute Of Technology,
Tumkur – 572216, Karnataka, India.
(sharayupradeep@gmail.com)

***Abstract* — The exponential growth of the Internet has led to a great deal of interest in developing useful and efficient tools and software to assist users in searching the web. Text is cheap, but the information i.e., knowing to which class a text belongs to, is expensive. Automatic categorization of text can provide this information at low cost, but the classifiers themselves must be built with expensive human effort, or trained from texts which have themselves been manually classified. Text classification is the process of classifying documents into predefined categories based on their content. Document retrieval, categorization and filtering can all be formulated as classification problem. Traditional information retrieval method use keywords occurring in documents to determine the class of the document. In this paper, we propose an association analysis approach for classifying the text using the generation of frequent item word sets (features), known as the Frequent-Pattern (FP) Growth. Naive Bayes classifier (Supervised classifier) is then used on derived features for final categorization.**

***Keywords* — *Supervised learning, Taxonomy, FP-growth, Naïve-Bayes classifier*.**

I. INTRODUCTION

Rapid advances in data collection and storage technology have enabled organizations to accumulate vast amounts of data. However, extracting useful information has proven extremely challenging. Often, traditional data analysis tools and techniques cannot be used because of the massive size of a data set. Sometimes, the non-traditional nature of the data means that traditional approaches cannot be applied even if the data set is relatively small.

Data mining is a collection of techniques for efficient automated discovery of previously unknown, valid, novel useful and understandable patterns in large databases. The patterns must be actionable so that they may be used in an enterprise's decision making or classification.

Available data for a mining session can be divided into three groups – *training data*, *test data* and *result validation data*. Training data are random samples of available data, used to develop a data

,

mining model. This model is tested for accuracy and conformity using the validation data.

A. Training set, test set, and validation set

A training set consisting of records whose class labels are known must be provided. The training set is used to build a classification model, which is subsequently applied to the test-set.

A test- set consisting of records with unknown class labels.

Using *validation set*, instead of using the training set to estimate the generalization error, the original training data is divided into two smaller subsets. One of the subsets is used for training, while the other known as *the validation set*, is used for estimating the generalization error.

B. Text taxonomy [1]

Text taxonomy [1] (or text classification) is the assignment of natural language documents to predefined categories according to their content. The set of categories is often called a "controlled vocabulary". The pre-defined categories are symbolic labels with no additional semantics.

Text taxonomy [1] is a kind of "supervised" learning where the categories are known beforehand and determined in advance for each training document. The text taxonomy process is illustrated in the figure-1. Documents pre-processing allows an efficient data manipulation and representation. Documents preprocessing techniques can be classified into Feature Extraction (FE) [1] and Feature Selection (FS) [1] approaches, as discussed below.

1) Feature Extraction [1]

The process of pre-processing is to make clear the border of each language structure and to eliminate as much as possible the language dependent factors, tokenization, stop words removal, and stemming. Feature extraction [1] is the fist step of pre processing which is used to presents the text

documents into clear word format. So removing stop words and stemming words is the pre-processing tasks. The documents in text taxonomy [1] are represented by a great amount of features and most of them could be irrelevant or noisy. Commonly the steeps taken place for the feature extractions (Figure-1) are:

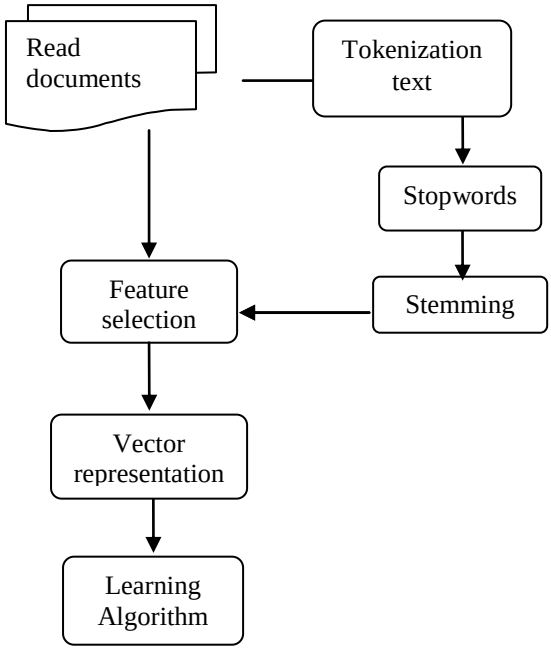

Figure-1. Text taxonomy process

Tokenization: A document is treated as a string, and then partitioned into a list of tokens.

Removing stop words: Stop words such as "the", "a", "and"… etc are frequently occurring, so the insignificant words need to be removed.

Stemming word: Applying the stemming algorithm that converts different word form into similar canonical form. This step is the process of conflating tokens to their root form, e.g. connection to connect, computing to compute etc.

2) Feature Selection [1]

After feature extraction [1] the important step in preprocessing of text classification, is feature selection to construct vector space, which improve the scalability, efficiency and accuracy of a text classifier. The main idea of feature selection is to select subset of features from the original documents. Feature selection [1] is performed by keeping the words with highest score according to predetermined measure of the importance of the word. The selected features retain original physical meaning and provide a better understanding for the data and learning process. For text classification a major problem is the high dimensionality of the feature space. Almost every text domain has much number of features, most of these features are not relevant and beneficial for text classification task, and even some noise features may sharply reduce the classification accuracy. Hence Feature selection [1] is commonly used in text classification to reduce the dimensionality of feature space and improve the efficiency and accuracy of classifiers.

This paper describes the FP-Growth technique and the Naïve Bayes classifier to categorize the text document on the basis of its content into its category.

II. RELATED WORK

This section briefly reviews related work on text taxonomy or classification. Text classification presents many challenges and difficulties. A number of methods have been discussed in the literature for text classification. Aurangzeb Khan, Baharum Baharudin, Lam Hong Lee*, Khairullah khan[1] provide a review of the theory and methods of document classification and text mining, focusing on the existing literature. Gabriel Fiol-Roig, Margaret Miro-Julia, and Eduardo Herraiz[2] analyzes the feasibility of an automatic web page classifier and proposes several classifiers and studies their precision. Rung-Ching Chen and Chung-Hsun Hsieh[3] proposes a web page classification method, which uses a support vector machine combining latent semantic analysis and web page feature selection. Y. H. Li and A.K. Jain[4] investigates four different methods for document classification: The Naïve Bayes classifier, the nearest neighbor classifier, decision trees and a subspace method. The results indicate that the Naïve Bayes classifier and the subspace method outperform the other two classifiers on their data sets. Ajay S Patil and B.V. Pawar[5] have attempted to classify web sites based on the content of their home pages using the Naïve Bayesian machine learning algorithm. S M Kamruzzaman and Chowdhury Mofizur Rahman[6] propose text categorization using words word relation i.e., association rules (Apriori algorithm) to derive feature set from pre-classified text documents. Naïve Bayes classifier is then used on derived features for final categorization. Jiawei Han, Jian Pei and Yiwen Yin[7] propose a novel frequent pattern tree (FP-tree) structure, which is an extended prefix tree structure for storing compressed, crucial information about frequent patterns, and develop an efficient FP-tree based mining method. Christian Borgelt[8] describes a C implementation of FP-growth algorithm, which contains two variants of the core operation of computing a projection of an FP-tree. Keyur J patel, Ketan J Savrvakar[9] propose the techniques for web page classification which includes Apriori Algorithm and implementation of Naive Bayes Classifier. T. Karthikeyan and N. Raviku[10] aims at giving a theoretical survey on some of the existing algorithms. S.Suriya, Dr.S.P.Shantharajah and R. Deepalakshmi[11] shows a complete survey of association rule mining in various domains. Margaret H. Dunham, Yongqiao Xiao, and Le Gruenwald, Zahid Hossain[12] provide an overview of association rule research. XindongWu, Vipin Kumar, J. Ross Quinlan[13] presents the top 10 data mining algorithms identified by the IEEE International Conference on Data Mining (ICDM) in December 2006: C4.5, *k*-Means, SVM, Apriori, EM, PageRank, AdaBoost, *k*NN, Naive Bayes, and CART. Fabrizio Sebastiani[14] discusses the main approaches to text categorization that fall within the machine learning paradigm. Raymond Kosala and Hendrik Blockeel[15] provide the survey of research in the area of Web mining, point out some confusions regarded the usage of the term Web mining and suggest three Web mining categories. S.Suriya, Dr.S.P.Shantharajah, and

R.Deepalakshmi[16] present a Complete Survey on Association Rule Mining with Relevance to Different Domain.

III. BACKGROUND STUDY

A. Data Mining[16]

Data mining [16] is a detailed process of analyzing large amounts of data and picking out the relevant information. It refers to extracting or mining knowledge from large amounts of data. It involves the following steps: cleaning and integrating data from data sources like databases, flatfiles, pre-treatment of selecting and transformation target data, mining the required knowledge and finally evaluation and presentation of knowledge. It is clearly explained pictorially in figure-2. In data mining [16], association rule learning is a most popular methodology to identify the interesting relations between the data stored in large databases.

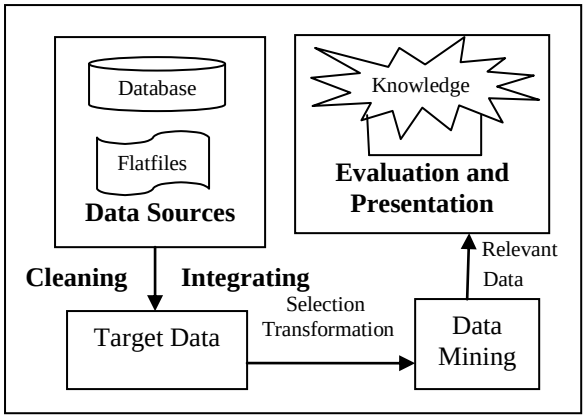

Figure-2. Data mining process

B. Association Rule [6][10][11][12][16]

Association rule mining [6][11][16] finds interesting association or correlation relationships among a large set of data items. In short association rule is based on associated relationships. The discovery of interesting association relationships among huge amounts of transaction records can help in many decision-making processes. Association rules are generated on the basis of two important terms namely minimum support threshold and minimum confidence threshold.

Let us consider the following assumptions to represent the association rule [6][10][12] in terms of mathematical representation, J = {i1, i2, . . . , im} be a set of items. Let D the task relevant data, be a set of database transactions where each transaction T is a set of items such that $T \subseteq J$. Each transaction is associated with an identifier, called TID. Let A be a set of items. A transaction T is said to contain A if and only if $A \subseteq T$. An association rule is an implication of the form A ==> B, where $A \subseteq J$, $B \subseteq J$, and $A \cap B = \Phi$. The rule A ==> B holds in the transaction set D with support s, where s is the percentage of transactions in D that contain $A \cup B$ (i.e. both A and B). This is taken to be the probability, $P(A \cup B)$. The rule A ==> B has confidence c in the transaction set t_d if c is the percentage of transaction in D containing A that also contain B. This is taken to be the conditional probability, P (B | A). That is,

$$support(A ==> B) = P(A \cup B) \text{ and}$$

$$confidence(A ==> B) = P(B \mid A).$$

Association Rules [12][16] that satisfy both a minimum support threshold and minimum confidence threshold are called strong association rules. A set of items is referred to as an itemset. In data mining research literature, "itemset" is more commonly used than "item set". An itemset that contains k items is a k-itemset. The occurrence frequency of an itemset is the number of transactions that contain the itemset. This is also known, simply as the frequency, support count, or count of the itemset. An itemset satisfies minimum support if the occurrence frequency of the itemset is greater than or equal to the product of minimum support and the total number of transactions in D. The number of transactions required for the itemset to satisfy minimum support is therefore referred to as the minimum support count. If an itemset satisfies minimum support, then it is a frequent itemset. The set of frequent k-itemsets is commonly denoted by L_k.

C. Vector space model [3]

The basic idea is to represent each document as a vector [2] of certain weighted word frequencies. In order to do so, the following parsing and extraction steps are needed.

1. Ignoring case, extract all unique words from the entire set of documents.
2. Eliminate non-content bearing "stopwords" such as "a", "and", "the", etc..
3. For each document, count the number of occurrences of each word.
4. Using heuristic or information-theoretic criteria, eliminate non-content-bearing "high-frequency" and "low-frequency" words.
5. After the above elimination, suppose *w* unique words remain. Assign a unique identifier between 1 and *w* to each remaining word, and a unique identifier between 1 and d to each document.

The above steps outline a simple preprocessing scheme.

D. FP-growth algorithm [7][8][13][15]

The design and construction of a frequent pattern tree is as follows: A frequent pattern tree [7] (or FP-tree in short) is a tree structure defined below.

1. It consists of one root labeled as "null", a set of item prefix subtrees as the children of the root, and a frequent-item header table.
2. Each node in the item prefix subtree consists of three fields: item-name, count, and node-link, where item-name registers which item this node represents, count registers the number of transactions represented by the portion of the path reaching this node, and node-link links to the next node in the FP-tree carrying the same item-name, or null if there is none.
3. Each entry in the frequent-item header table consists of two fields, (1) item-name and (2) head of node-link, which points to the first node in the FP-tree [7][8] carrying the item-name. Based on this definition, we have the following FP-tree construction algorithm.

Algorithm 1 (FP-tree construction):

Input: A transaction database D and a minimum support threshold ξ.

Output: The FP-Tree [8] frequent pattern tree of D.
Method: The FP-tree is constructed in the following steps.

1. Collect the set of frequent items (F_{items}) and their support count after scanning the transaction database D once. Sort F_{items} according to descending support count as L_{freq}, the list of frequent items.
2. Create a root of an FP-tree, and label it as "null". For each transaction I_{Trans} in D do the following.

Select and sort the frequent items in I_{Trans} according to the order of L_{freq}. Let the sorted frequent item list in I_{Trans} be [e | E_{list}], where e is the first element and E_{list} is the remaining list. Call insert_tree([e | E_{list}], T), which is performed as follows.

Procedure insert_tree([e | E_{list}], T)
If T has a child N such that N.item-name = e.item-name, then increment N's count by 1; else create a new node N, and let its count be 1, its parent link be linked to T, and its node link to the nodes with the same item-name via the node-link structure. If E_{list} is nonempty, call insert_tree(E_{list}, N) recursively.

Algorithm 2 (FP-growth : Mining frequent patterns with FP-tree and by pattern fragment growth).

Input: A database D, represented by FP-tree constructed based on Algorithm 1, and a minimum support threshold ξ.
Output: The complete set of frequent patterns.
Method: Call FP-growth (FP-tree, null), which is implemented as follows.

Procedure FP-growth(Tree, α) {
If Tree contains a single prefix path then {
Let P be the single prefix-path part of Tree;
Let Q be the multipath part with the top branching node replaced by a null root;
for each combination (denoted as β) of the nodes in the path P do
generate pattern β U α with support = minimum support of nodes in β;
Let freq_pattern_set(P) be the set of patterns so generated;}
else let Q be tree;
for each item a_i in Q do {
generate pattern β = a_i U α with support = a_i.support;
Construct β's conditional pattern-base and then β's conditional FP-tree $Tree_{\beta;}$
If $tree_\beta \neq \Phi$ then call FP-growth($Tree_{\beta,}$ β);
Let freq_pattern_set(Q) be the set of patterns so generated;}
return(freq_pattern_set(P) U freq_pattern_set(Q) U (freq_pattern_set(P) * freq_pattern_set(Q)))
}

E. Naïve Bayes classifier [2][5][6][9][13]

Bayesian classification [5] is based on Bayes theorem. Bayesian classifiers have also exhibited high accuracy and speed when applied to large database. Naïve Bayes classifier [6][9] assumes that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called *class conditional independence*. It is made to simplify the computations involved and, in this sense, is considered "naïve".

While applying Naïve Bayes classifier [6] to classify text, each word position in a document is defined as an attribute and the value of that attribute to be the word found in that position. Here Naïve Bayes classification can be given by:

$$VNB = \text{argmax } P(Vj) \Pi P(aj \mid Vj) \quad ...(Eq.\ 1)$$

Here VNB is the classification that maximizes the probability of observing the words that were actually found in the example documents, subject to the usual Naïve Bayes independence assumption. The first term can be estimated based on the fraction of each class in the training data. The following equation is used for estimating the second term:

$$\frac{n_k + 1}{n + |\text{ vocubulary }|} \quad ...(Eq.\ 2)$$

where n is the total number of word positions in all training examples whose target value is Vj, n_k is the number of items that word is found among these n word positions, and | vocubulary | is the total number of distinct words found within the training data.

IV. PROPOSED METHOD

1. *Pre-processing Training Data Set* (*TRD*) *and Test Data Set* (*TED*):
   - Remove the stop and unwanted words from both *TRD* and *TED*.
   - Select noun as the keywords from both data set and remove duplicate keywords from each document.
   - Do stemming using porter algorithm on both data sets.
   - Save each processed *n* pages of *TRD* as document *Dk* , where k = 1, 2, 3,…, *n* and each processed *m* pages of *TED* as document *TEDj*, where j = 1, 2, 3,…, *m*.
2. *Create term document matrix:*
   Term document matrix, *T*, is created by counting the number of occurrences of each term in each document D_k. Each row t_i of *T* shows a term's occurrence in each document D_k.
3. *Extraction of frequent sets:*
   FP-growth algorithm is used to generate frequent word sets from the term document matrix *T* using the value of minimum support, *min_sup*, given as an input and stored in *F*. Calculate the probability values of each frequent word sets stored in *F* using Naïve Bayes method.
4. *Finding matching word set(s):*
   Search for matching word set(s) or its subset (containing items more than one) in the list of word sets collected from *F* with that of subset(s) of word of new test document using regular expression search.
5. *Calculate the probability values of target class:*
   a. Collect the corresponding probability values of matched word set(s) for each target class.
   b. Calculate the probability values for each target class from naïve based classification approach using Eq. 1 and Eq. 2.
6. *Assignment the new document to that target class which has highest probability values*.

V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a text (plain text document) taxonomy using a supervised learning approach. Here, we have used the FP-growth technique combined with the Naïve Bayes classifier to classify the text document. The purpose of FP-growth is to generate frequent word sets from training data set. We have considered text documents as transactions and the set of frequently occurring words as a set of items in the transaction. The new documents are classified by applying the Naïve Bayes classifier on these derived sets. It categorizes the text into very broad categories. The results are quite encouraging. This approach can be used by search engines for effective categorization of website to build an automated website directory based on type of organization. However in this experiment, only distinct and non hierarchical categories are considered. The same algorithm cold also be used to classify the documents into more specific categories(hierarchical classification) by changing the feature set.

REFERENCES

[1] Aurangzeb Khan, Baharum Baharudin, Lam Hong Lee*, Khairullah khan, "A Review of Machine Learning Algorithms for Text-Documents Classification", JOAIT, Vol 1, 2010, 4-20.

[2] Gabriel Fiol-Roig, Margaret Miro-Julia, Eduardo Herraiz, "Data mining techniques for web page classification".

[3] Rung-Ching Chen, Chung-Hsun Hsieh , "Web page classification based on a support vector machine using a weighted vote schema", Expert systems with applications 31 (2006) 427-435.

[4] Y. H. Li and A.K. Jain, "Classification of text documents", The computer journal, Vol. 14, No.8, 1998, 537-546.

[5] Ajay S Patil and B.V. Pawar, "Automated classification of web sites using Naïve Bayesian algorithm", IMECS, Vol. I, 2012, March 14 – 16.

[6] S M Kamruzzaman and Chowdhury Mofizur Rahman., "Text categorization using Association Rule and Naïve Bayes classifier", Asian Journal of Information Technology, Vol. 3, No. 9, pp 657-665, Sep. 2004.

[7] Jiawei Han, Jian Pei and Yiwen Yin, "Mining frequent patterns without candidate generation",SIGMOD-2000, Paper-ID:196, 1-11.

[8] Christian Borgelt, "An implementation of the FP-growth algorithm", Proceedings of OSDM'05, 1-5.

[9] Keyur J patel, Ketan J Savrvakar, "Web-classification using data mining", IJARCCE, Vol.2, Issue 7, 2013,2513-2520.

[10] T. Karthikeyan and N. Ravikumar, "A Survey on Association Rule Mining", IJARCCE, Vol. 3, Issue 1, 2014, 5223-5227.

[11] S.Suriya, Dr.S.P.Shantharajah, R.Deepalakshmi, "A Complete Survey on Association Rule Mining with Relevance to Different Domain", IJASTR, Issue 2, Vol 1, 2012, 163-168.

[12] Margaret H. Dunham, Yongqiao Xiao, Le Gruenwald, Zahid Hossain, "A Survey of association rules".

[13] XindongWu, Vipin Kumar, J. Ross Quinlan, "Top 10 algorithms in data mining", Springer-Verlag London Limited 2007, 1-37.

[14] Fabrizio Sebastiani, "Machine Learning in Automated Text Categorization", ACM-CSUR, Vol. 34, Issue 1, 2002, 1-47.

[15] Raymond Kosala and Hendrik Blockeel, "Web Mining Research: A Survey", ACM SIGKDD Newsletter, Vol 2, Issue 1, 2000, 1-15.

[16] S.Suriya, Dr.S.P.Shantharajah, and R.Deepalakshmi, " A Complete Survey on Association Rule Mining with Relevance to Different Domain", IJASTR, Issue 2, Vol. 1, 2010, 4-20.