Recent Trends in Text Classification Techniques

Nidhi
M.E, Computer Science & Engineering
University Institute of Engineering & Technology,
Panjab University, Chandigarh

Vishal Gupta
Assistant Professor, Computer Science & Engineering Department
University Institute of Engineering & Technology,
Panjab University, Chandigarh

ABSTRACT
Text Mining is the discovery of valuable, yet hidden, information from the text document. Text classification (Also called Text Categorization) is one of the important research issues in the field of text mining. With the dramatic increase in the amount of content available in digital forms gives rise to a problem to manage this online textual data. As a result, it has become a necessary to classify/categorize large texts (documents) into specific classes. Text Classification assigns a text document to one of a set of predefined classes. This paper covers different text classification techniques and also includes Classifier Architecture and Text Classification Applications.

Terms
Text Mining, Text Classification, Applications, Classifier Architecture, Classification Techniques.

Keywords
KNN, Naïve Bayes, Support Vector Machine, Term Graph Model, Association Based Classification, Decision Tree Induction, Centroid based classification, Classification using neural network.

1. INTRODUCTION
Text Mining [1] [2] refers to the process of deriving high-quality information from text. 'High quality' in text mining means that information extracted should be relevant to the user, and according to the interest of the user. The text document may be a plain text document (e.g. ASCII) or a tagged text document (e.g. HTML/XML). And Typical text mining tasks include text classification (text categorization), text clustering, concept/entity extraction, topic tracking, information visualization, question answering, document summarization etc.

Text (or Document) classification is an active research area of text mining, where the documents are classified into predefined classes. Mostly-text documents include letters, newspapers, articles, blogs, technical reports, proceedings, and journal papers, etc. Document Filtering, also based on the classification algorithm to extract the relevant documents related to specific topic from the set of documents [3].

Text Classification tasks can be divided into two sorts: supervised document classification where some external mechanism (such as human feedback) provides information on the correct classification for documents or to define classes for the classifier, and unsupervised document classification (also known as document clustering), where the classification must be done without any external reference, this system do not have predefined classes. There is also another task called semi-supervised document classification, where some documents are labeled by the external mechanism (means some documents are already classified for better learning of the classifier).

Need for Automatic Text Classification: To classify millions of text document manually is an expensive and time consuming task. Therefore, automatic text classifier is constructed using pre-classified sample documents whose accuracy and time efficiency is much better than manual text classification. In this paper we summarize text classification techniques that are used to classify the text documents into predefined classes [4].

Section 2 and 3 defines text classification applications and classifier Architecture respectively. And in Section 4, we summarized different classification techniques.

2. CLASSIFICATION APPLICATIONS
Text (document) classification is an important task in document processing. The applications of classification are:

2.1 Automated authorship attribution
Authorship attribution is the science of determining the author of a text document, from a predefined set of candidate authors or inferring the characteristic of the author from the characteristics of documents written by that author [5].

2.2 Automatic Document Distribution
Text classification also allows the efficient automatic distribution of documents via email or fax by eliminating the time consuming, manual process of faxing or mailing. And this can be achieved by first classifying the documents according to sender and message type [5].

2.3 Automated survey coding
Survey coding is the task of assigning a symbolic code from a predefined set of such codes to the answer that a person has given in response to an open-ended question in a questionnaire (survey) [5]. Survey coding has several applications, especially in the social sciences, ranging from the simple classification of respondents on the basis of their answers to the extraction of statistics on political opinions, health, and customer satisfaction etc.

2.4 Word sense disambiguation
Word sense disambiguation (WSD) is the activity of finding, given the occurrence in a text of an ambiguous (i.e. polysemous or homonymous) word, the sense of this particular word occurrence. For instance, bank may have (at least) two different senses in English, as in the Bank of England (a financial institution) or the bank of river Thames [5].

2.5 Text filtering
Text filtering is for example the activity to classify text document containing specific keywords or several keywords. Typical cases of filtering systems are e-mail filters, newsfeed filters [5].

2.6 Document clustering
Document clustering (also called unsupervised learning) is the act of collecting similar documents into clusters. In this, we do not have any external sources to provide information on the correct clustering for documents [5].

Text classification also improves indexing efficiency in Digital Libraries by classifying the documents into different categories like sports, science, mathematics etc. Other applications are, topic spotting, automatically determining the topic of a text; language guessing, automatically determining the language of a text [5].

3. CLASSIFIER ARCHITECTURE
Text classification is a fundamental task in document processing. The goal of text classification is to classify a set of documents into a fixed number of predefined categories/classes. A document may belong to more than one class. When classifying a document, a document is represented as a "bag of words". It does not attempt to process the actual information as information extraction does. Rather, in simple text classification task, it only counts words (term frequency) that appear and, from the count, identifies the main topics that the document covers e.g. if in the document, cricket word comes frequently then "cricket" is assigned as its topic (or class) [6] [7]. Classification is a two step process. First step is Model Construction.

Model Construction: Also called Training Phase or Learning Phase; the set of documents used for model construction is called training set. It describes a set of predetermined classes. Each document/sample in the training set is assumed to belong to a predefined class (labeled documents). The model is represented as classification rules, decision trees, or mathematical formulae [7] [8].

Mode Usage: This is the 2nd step in classification. Also called Testing Phase or Classification Phase, it is used for classifying future or unlabeled documents. The known label of test document/sample is compared with the classified result to estimate the accuracy of the classifier. For e.g. the labeled documents of the training set, is used further to classify unlabeled documents. Test set is independent of training set [6] [7] [8].

Figure 1 shows the overall flow diagram of the text classification task. Consider a set of labeled documents from a source $D = [d_1, d_2, ..., d_n]$ belonging to a set of classes $C = [c_1, c_2, ..., c_p]$. The text classification task is to train the classifier using these labeled documents, and assign categories/classes to the new, unlabeled documents. In the training phase, the n documents are arranged in p separate folders, where each folder corresponds to one class. In the next step, the training data set is prepared via a feature selection process [6] [7].

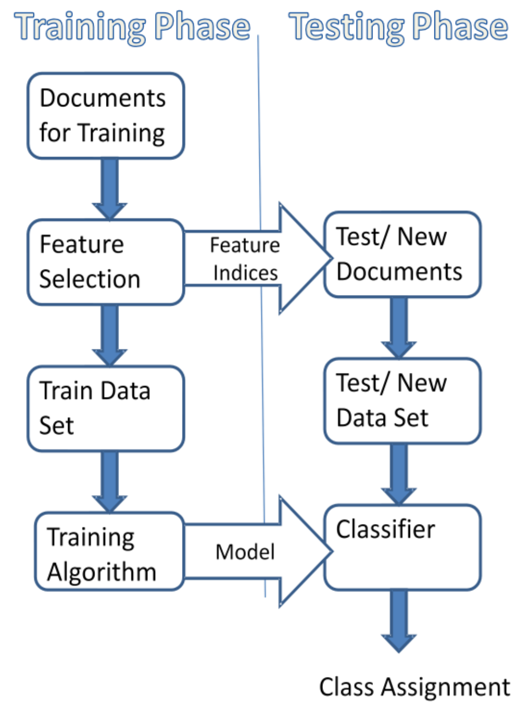


Figure 1: Flow Diagram of Text Classification

Any classifier is unable to understand a document in its raw format; a document has to be converted into a standard representation. It is observed from previous research that words work well as features for many text classification techniques [4] [6]. In the feature space representation, the text documents are represented as sequence of words called "Bag of Words". Bag of Words (BoW) is one of the basic methods of representing a document. The BoW is used to form a vector representing a document, with one component in the vector for each word in the document. These components are computed using term frequency (TF). Such representation of a set of documents as a vector is known as Vector Space Model [9]. Text can also be tokenized using inverse document frequency (IDF), term frequency inverse document frequency (TFIDF) or using binary representation [6].

Text classification also presents many challenges and difficulties. First, it is difficult to capture high level semantics from a few key words. Second, high dimensionality (thousands of features) and variable length of the documents, place both efficiency and accuracy demands on classification systems [9].

4. CLASSIFICATION TECHNIQUES

There are many techniques which are used for text classification. Following are some techniques:

- Nearest Neighbour classifier
- Bayesian Classification
- Support Vector Machine
- Association Based Classification
- Term Graph Model
- Centroid Based Classification
- Decision Tree Induction
- Classification using Neural Network

4.1 Nearest Neighbour Classifier

The K-nearest neighbors algorithm (KNN) [4] [8] [9] [10] [11] is based on closest training examples. The KNN algorithm is simple, valid and non-parameter method. KNN is also called instance-based learning or lazy learning. In this, each document is represented by nodes. For classification, distance between each labeled node (labeled document) and unlabeled node (unlabeled document) is calculated. And to decide whether the document d_i belongs to class C_k, the similarity (d_i, d_j) or dissimilarity (d_i, d_j) to all documents in the training set is determined. If there are N labeled nodes, the complexity of finding the label for an unlabeled node from KNN is $N * \log k$. The document is simply assigned to the class of its nearest neighbor, if $k = 1$. The performance of the classifier depends only on two parameters, k and similarity or dissimilarity value.

Advantages: It is simple to implement as it needs only two parameters, and is robust to noisy training data. KNN is does not do any learning, it simply stores all the training examples. KNN also well suited for multi-modal classes as its classification decision is based on a small neighborhood of similar documents (i.e. the major class).

Limitations: The time needed to compute similarity/dissimilarity is huge. In practical, it is impossible to implement KNN algorithm for high dimensions and huge samples. As a result, Classification cost becomes very high for the Nearest-Neighbor. Also, the classifier grows with the number of training documents.

A new approach called TFKNN (Tree-Fast-K-Nearest-Neighbor) presented in [10], search the exact k nearest neighbors quickly. With this new approach, the searching scope is reduced and the time of similarity computing is decreased largely.

4.2 Bayesian Classification

The Naïve Bayes Classifier [4] [9] [13] is the simplest probabilistic classifier used to classify the text documents. It severe assumption that each feature word is independent of other feature words in a document. The basic idea is to use the joint probabilities of words and categories to estimate the class of a given document. Given a document d_i, the probability with each class c_j is calculated as

$$P(c_j \mid d_i) = \frac{P(d_i \mid c_j)}{P(d_i)} P(c_j)$$

As $P(d_i)$ is the same for all class, then $\mathrm{label}(d_i)$ the class (or label) of d_i, can be determined by

$$\mathrm{label}(d_i) = \arg\mathrm{Max}_{c_j} \{ P(c_j | d_i) \} = \arg\mathrm{Max}_{c_j} \{ P(d_i | c_j) \cdot P(c_j) \}$$

For text classification there are two different models of Naïve Bayes classifiers: Multi-Variate Bernoulli Event Model and the Multinomial Event Model. The idea of calculating the probability $P(d_i | c_j)$ in above equation is different in these two models. A document d_i is represented with the vector of $|V|$ dimensions i.e. $(t_1, t_2, ..., t_{|V|})$. In the multi-variate Bernoulli event model, a vocabulary V is given, and the sequence of the words is considered. The k^{th} dimension of the vector corresponds to word w_k from V and its value is either 1 (if word w_k present in the document) or 0 (otherwise). In the multinomial event model, no order of the words is considered, and the value of word w_k is calculated as the frequency of that word in the document.

Naïve Bayes classifier is highly sensitive to feature selection. So the study of feature evaluation metrics for it is very necessary, and two effective metrics for the Naïve Bayes classifier applied on multi-class datasets: Multi-class Odds Ratio (MOR) and Class Discriminating Measure (CDM).

Advantages: The Naive Bayes classifier is a popular machine learning method for text classification because it is fast and easy to implement and performs well.

Limitations: Naive-Bayes is manageable only for low dimensions. Its assumption makes higher efficiency possible but this adversely affects the quality of the results, if feature words are interrelated.

4.3 Support Vector Machine

The Support Vector Machine (SVM) [4] [14] [15] technique is a popular and highly accurate machine learning method for classification problems. SVM try to find an optimal hyperplane within the input space so as to correctly classify the binary (or multi-class) classification problem. For linearly separable space (i.e for binary classification problem), the hyperplane is written as

$$w \cdot x + b = 0$$

Here x is an arbitrary object to be classified; the vector w and constant b are learned from a training set of linearly separable objects. In case of linearly separable data, SVM separates the positive and negative training examples with a maximum margin.

Advantages: Support Vector Machines are less susceptible to overfitting than other learning methods since the model complexity is independent of the feature space dimension. SVM-based approaches can handle large feature spaces with excellent classification accuracy. It produces the best results both on test and training sets and is robust with respect to the number of features and very fast at training and at classification.

Limitations: The complexity in the implementation. And it cannot scale well with the number of documents in the text collections.

4.4 Association based Classification

Classification based on associations (CBA) [16] [17] [18] [19] [20] integrates classification and association rule mining. It generates class association rules and does classification more accurately than decision tree, C4.5. Classification association rules (CARs) are association rules with the class on the right hand side of the rules and conditions on the left side of the rules. These rules are extracted from the available training data and the most adequate rules are selected to build an "associative classification model".

In text classification, CBA is used to classify text documents into topic hierarchies. And rules are extracted using Apriori Algorithm. So when CBA does classification, more than one rule can fit a certain case and the final class will be derived from the rule with highest confidence (or support).

Instead of relying on a single rule for classification, Classification based on Multiple Association Rules (CMAR) determines the class label by a set of rules, taking into account that the rule with highest confident (or support) might not always produce best result to classify data. Classification based on Multiple Association Rules, extends the FP-growth frequent pattern mining algorithm to obtain and store rules for classification. This improves the speed, efficiency and accuracy of the classification model.

CPAR, Classification based on Predictive Association Rules, generates a much smaller set of high-quality predictive rules directly from the training data and avoid generating redundant rules in comparison with associative classification and CMAR. As a result, CPAR is much more time-efficient in both rule generation and class prediction.

Advantages: Associative classification has high classification accuracy and strong flexibility at handling textual data. The rules generated are also used for comparing the quality of different association rule mining approaches.

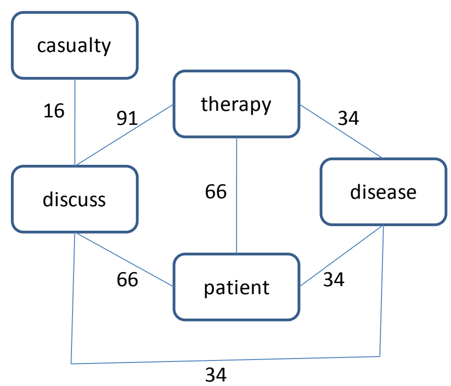
Limitations: CBA results in huge set of mined rules. It becomes challenging to store, retrieve, prune, and sort a large number of rules efficiently for classification. And sometimes it suffers from biased classification or overfitting problem too since the classification is based on single high-confidence (or support) rule.

4.5 Term Graph Model

Most existing text classification methods are based on representing the documents using the vector space model but due to this, sometimes, important information, such as the relationship among words, is lost. The term graph model [21] [22] [23] is an improved version of the vector space model. It represents not only the content of a document but also the relationship among words. A graph model is built to represent all extracted relations. To constructs graph, first, we construct a node for each unique term that appears at least once in the frequent datasets. Then we create edges between two nodes u and v if and only if they are both contained in one frequent dataset. The weight of the edge between u and v is the largest support value among all the frequent datasets that contains both of them.

2a) Table 1. Frequent Datasets

Dataset	Support
{therapy, discuss}	91
{therapy, discuss, patient}	66
{therapy, discuss, patient, disease}	34
{casualty, discuss}	16



2b) The corresponding graph

Figure 2: an example Term Graph

In the above table 2(a), it consists of frequent datasets having support greater than minimum support. And in figure 2(b), on the basis of fig. 2(a), relationships among words are shown.

Given a relation $t = (e_i; r_{ij}; e_j)$, where e_i and e_j are considered as vertices in the graph and r_{ij} is considered as the edge connecting e_i and e_j. The weight of the edge w_{ij} is calculated based on the importance of r_{ij} in its documents.

For text classification, we consider two different approaches, first is, Ranking Term, word that appears frequently with other words in the text collections is considered to be an important word. Second is Rank Correlation Coefficient, to calculate the similarity between a document and a class.

Advantages: The graph representation of the document is more expressive than standard bag of words representation, and consequently gives improved classification accuracy. We can also preserve and extract the hidden relationships among terms in the documents.

Limitations: The computational complexity of the graph representation for text classification is the main disadvantage of the approach.

4.6 Centroid Based Classification

Due to simplicity and linearity, for text classification Centroid Classifier [4] [24] [25] has become a commonly used method. Its basic idea is to construct a prototype vector, or centroid, per class using training documents. In terms of accuracy, K-nearest neighbor algorithm usually performs well but it is slow in recognition phase because the distances/similarities between the new data point to be recognized and all the training data need to be computed. On the other hand, centroid-based classification algorithms are very fast, because only as many distance/similarity computations as the number of centroids (i.e. classes) needs to be done.

Centroid Classifier has proved to be a simple and efficient method. In this approach to improve the performance of centroid-based classifier normalization is an important factor when documents in text collection are of different sizes and/or the numbers of documents in classes are unbalanced. The classification task is to find the most similar class (with cosine similarity) to the vector of the document we would like to classify. Based on the documents in each class the centroid based classifier selects a single representative called centroid and then it works like KNN classifier with $k = 1$.

Advantages: The technique is simple to implement and flexible to text data. It has relatively less computation than other methods in both the learning and classification stages.

Limitations: Centroid Classifier makes a simple hypothesis that a given document should be assigned a particular class if the similarity of the given document to the centroid of the class is the largest. But this supposition is often violated when a document from class A sharing more similarity with the centroid of class B than that of class A which lead to poor performance of the classification. Therefore, to enhance the performance of Centroid Classifier, take advantage of training errors to successively update the classification model by batch.

4.7 Decision Tree Induction

Decision trees [7] [8] are the most widely used inductive learning methods. Decision tree classification is the learning of decision trees from labeled training documents. One of the most well known decision tree algorithms is ID3 and its successor C4.5 and C5. A decision tree is a flowchart like tree structures, where each internal node denotes a test on document, each branch represents an outcome of the test, and each leaf node holds a class label. It is a top-down method which recursively constructs a decision tree classifier.

Advantages: Their robustness to noisy data and their capability to learn disjunctive expressions seem suitable for document classification. Decision trees are simple to understand and interpret. They require little data and are able to handle both numerical and categorical data. This algorithm scales well, even where there are varying numbers of training examples.

Limitations: Decision-tree learning algorithms are based on heuristic algorithms such as the greedy algorithm where decisions are made at each node locally. Such algorithms cannot guarantee to return the globally optimal decision tree. A fully grown tree may be prone to overfitting, as some branches may be too specific to the training data. Most Decision Tree learning methods thus include a method for growing the tree and one for pruning it, for removing the overly specific branches.

4.8 Classification using Neural Network

Neural networks [26] [27] [28] [29] have emerged as an important tool for classification. One major limitation of the statistical models (e.g. Naïve Bayes) is that they work well only when the underlying assumptions are satisfied. But neural networks are data driven self-adaptive methods in that they can adjust themselves to the data without any explicit specification of functional or distributional form for the underlying model. Applications include bankruptcy prediction, handwriting recognition, speech recognition, fault detection, medical diagnosis etc. For classifying a given test document d_i , its term weights w_{ki} are loaded into the input units; the activation of these units is propagated forward through the network, and the value of the output unit(s) determines the categorization decision(s).

The basic design of the three layered back-propagation neural network as shown in figure 3, consist of m inputs on input layer, m outputs on output layer and k neurons on the hidden layer, where $k < m$, under the assumption that the documents are represented in an m-dimensional space. Each input node is connected to all Hidden nodes. And each hidden node is connected to all of the output nodes. The number of output nodes is equal to the defined classes for the classification. The network is then trained, under standard back propagation method that requires set of training documents and predefined classes.

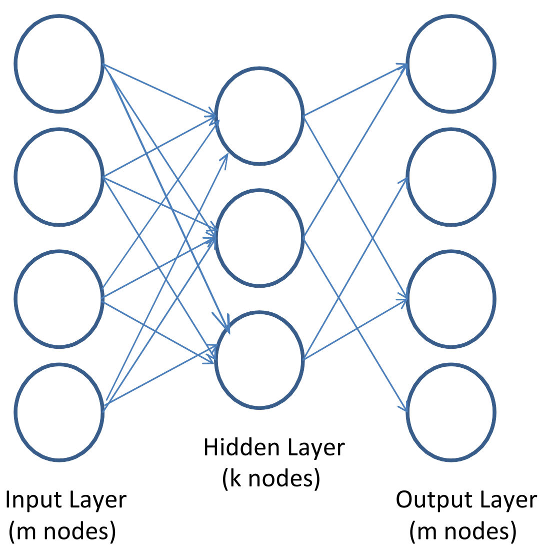


Figure 3: Typical three layered back-propagation neural network.

Advantages: Neural networks are nonlinear models, which makes them flexible in modeling real world complex relationships. Neural networks are able to estimate the posterior probabilities, which provide the basis for establishing classification rule and performing statistical analysis. More than two hidden nodes provide better classification.

Limitation: With increase in the number of input and hidden nodes, the parameters needed for neural network also increases this result in overfitting of the data.

5. CONCLUSION

With the dramatic rise in the use of the internet, there has been an explosion in the volume of online documents. Text Classification (Text Categorization), the assignment of text documents to one or more predefined classes based on their content, is an important component in many information management tasks.

K-nearest neighbors algorithm (KNN) is the simplest method for deciding the class of the unlabeled documents and is a popular non-parametric method. But for the high dimensions, its computational time increases as a result this method is not suitable for such documents. SVMs and Neural Network tend to perform much better when dealing with multi dimensions.

For SVMs and Neural Network, large sample size is required to achieve maximum accuracy of the classifier. Hence, we need large storage space for both training and testing documents. Where as Naïve Bayes may need a relatively less dataset and require little storage space.

Naïve Bayes has high bias and does not produce efficient results if words (or terms) are correlated, in order to maintain the correlation among adjacent words, Term Graph Model is preferred. KNN, Neural Network is generally considered intolerant of noise; where association based classification and decision trees are considered resistant to noise because their pruning strategies avoid overfitting the noisy data.

Compare to other classifiers, SVM performs well, its accuracy, speed of learning, speed of classification, tolerance to irrelevant features and noisy data is much better than other classifiers. Also, one remarkable property of SVMs is that their ability to learn can be independent of the dimensionality of the feature space. But still it is difficult to recommend any one technique as superior to others as the choice of a modeling technique depends on organizational requirements and the data on hand.

6. REFERENCES

[1] J.H. Kroeze, M.C. Matthee and T.J.D. Bothma, July 2007, “Differentiating between data-mining and text-mining terminology”, “doi: 10.1.1.95.7062”.

[2] F. Sebastiani, 2002 “Machine learning in automated text categorization”, ACM Computer Surveys 34(1), 1–47.

[3] Nawei Chen and Dorothea Blostein, 2006, “A survey of document image classification: problem statement, classifier architecture and performance evaluation”, Springer-Verlag, “doi: 10.1007/s10032-006-0020-2”.

[4] Christoph Goller, Joachim Löning, Thilo Will and Werner Wolff, 2009, “Automatic Document Classification: A thorough Evaluation of various Methods”, “doi=10.1.1.90.966”.

[5] Fabrizio Sebastiani, 2005 “Text categorization”, In Alessandro Zanasi (ed.), Text Mining and its Applications, WIT Press, Southampton, UK, 2005, pp. 109-129.

[6] Vishal Gupta, Gurpreet S. Lehal, August 2009 “A Survey of Text Mining Techniques and Applications”, Journal of Emerging Technologies in Web Intelligence, VOL. 1, NO. 1.

[7] Jiawei Han, Michelin Kamber, 2001, “Data Mining Concepts and Techniques”, Morgan Kaufmann publishers, USA, 70-181.

[8] Megha Gupta, Naveen Aggrawal, 19-20 March 2010, “Classification Techniques Analysis”, NCCI 2010 - National Conference on Computational Instrumentation CSIO Chandigarh, INDIA, pp. 128-131.

[9] B S Harish, D S Guru and S Manjunath, 2010, “Representation and Classification of Text Documents: A Brief Review”, IJCA Special Issue on “Recent Trends in Image Processing and Pattern Recognition” RTIPPR.

[10] Yu Wang and Zheng-Ou Wang, 2007, “ A Fast KNN Algorithm for Text Classification”, Machine Learning and Cybernetics, International Conference on, Vol. 6, pp. 3436-3441, doi : 10.1109/ICMLC.2007.4370742, Hong Kong, IEEE.

[11] Wei Wang, Sujian Li and Chen Wang, 2008, “ICL at NTCIR-7: An Improved KNN Algorithm for Text Categorization”, Proceedings of NTCIR-7 Workshop Meeting, December 16–19, Tokyo, Japan.

[12] Yancong Zhou and Hyuk Cho, 2001, “Classification Algorithms on Text Documents”.

[13] Jingnian Chen, Houkuan Huang, Shengfeng Tian and Youli Qu, 2009, “Feature selection for text classification with Naïve Bayes”, Expert Systems with Applications: An International Journal, Volume 36 Issue 3, Elsevier.

[14] Wen Zhang, Taketoshi Yoshida and Xijin Tang, 2008, “Text classification based on multi-word with support vector machine”, Journal: Knowledge Based Systems - KBS , vol. 21, no. 8, pp. 879-886, doi: 10.1016/j.knosys.2008.03.044, Elsevier.

[15] Steve R. Gunn, 1998, “Support Vector Machines for Classification and Regression”, University of Southampton.

[16] Wenmin Li, Jiawei Han and Jian Pei, 2001, “CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules”, IEEE International Conference on Data Mining - ICDM , pp. 369-376, DOI: 10.1109/ICDM.2001.989541.

[17] Xiaoxin Yin, Jiawei Han. CPAR, 2003,” Classification based on Predictive Association Rules”, in Proceedings of SDM, doi=10.1.1.12.7268.

[18] Fernando Berzal, Juan-Carlos Cubero, Nicolás Marín, Daniel Sánchez, Jose-María Serrano, Amparo Vila, *“Association rule evaluation for classification purposes”*.

[19] Chowdhury Mofizur Rahman, Ferdous Ahmed Sohel, Parvez Naushad, S. M. Kamruzzaman, May 2003, “Text Classification using the Concept of Association Rule of

Data Mining", International Conference on Information Technology, Kathmandu, Nepal, pp. 234-241.

[20] Xin Lu, Barbara Di Eugenio, Stellan Ohlsson, 2007, "Learning Tutorial Rules Using Classification Based On Associations" , In Proceeding of the 2007 conference on Artificial Intelligence in Education, ISBN: 978-1-58603-764-2.

[21] Wei Wang , Diep Bich Do , Xuemin Lin, 2005, "Term Graph Model for Text Classification", doi=10.1.1.149.6207.

[22] Chuntao Jiang, Frans Coenen, Robert Sanderson, Michele Zito, May 2010, "Text classification using graph mining-based feature extraction", Journal Knowledge-Based Systems Volume 23 Issue 4, Elsevier.

[23] Dat Huynh, Dat Tran, Wanli Ma, Dharmendra Sharma, 2011, "A New Term Ranking Method Based on Relation Extraction and Graph Model for Text Classification", Faculty of Information Sciences and Engineering, University of Canberra ACT 2601, Australia.

[24] Songbo Tan, 2008, "An improved centroid classifier for text categorization", Expert Systems with Applications 35, 279–285, Elsevier.

[25] Verayuth Lertnattee, Thanaruk Theeramunkong, 2006, "Class normalization in centroid-based text categorization", Information Sciences 176, 1712–1738, Elsevier.

[26] Guoqiang Peter Zhang, November 2000, "Neural Networks for Classification: A Survey", IEEE Transactions on systems, man and cybernetics-Part C, Applications and Reviews, Vol. 30, NO. 4.

[27] Larry Manevitz, Malik Yousef, 2007, "One-class document classification via Neural Networks", Neurocomputing 70, 1466–1481, Elsevier.

[28] David Faraggi, Richard Simon, 1995, "The maximum likelihood neural network as a statistical classification model", Journal of Statistical Planning and Inference 46, 93-104, Elsevier.

[29] Ali Selamat, Sigeru Omatu, 2004, "Web page feature selection and classification using neural networks", Information Sciences 158, 69–88, Elsevier.