International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 9 (2018) pp. 6536-6544

Two Dimensional Feature Extraction and Blog Classification using Artificial Neural Network

Aruna Devi K

Research Scholar, Mother Teresa Women's University, Kodaikanal 624102, India.
Assistant Professor in Computer Science (PG), Kristu Jayanti College, Bengaluru 560077, India.

Kathirvalavakumar T

Research Center in Computer Science, V.H.N.Senthikumara Nadar College, Virudhunagar 626001, India.

Abstract

A two dimensional feature reduction for automatic blog classification using artificial neural network is proposed. A blog can be manually categorized using the tags provided in the page itself. In spite of the tag, the blog contents can diverge to some other topic as it can be written by novice content writers also. In the proposed method the contents are represented as a collection of features. Significant features are identified using term weighting technique and information gain in the first phase. The leader algorithm selects only representatives of the clusters in the next phase, thus minimizes the number of patterns from the large amount of blogs. The selected patterns are fed to ANN classifier for training that scale down the training time of the classifier and also lead to better performance. The proposed method is implemented and validated on a live dataset. The results outperform the existing methods in terms of accuracy and training time.

Keywords: Blog Classification; Term Weighting; Information Gain; Leader Algorithm; Artificial Neural Networks; Dimensionality Reduction

INTRODUCTION

The Web infrastructure is flooded with augmented structured and unstructured data in various domains. With the rapid growth of contents in the web it is hard to leverage the maximum useful information appropriate to the user's requirement on a specific domain. A directory is a powerful tool that automatically categorizes the web pages and blogs under different classes of interest. A Blog is kind of website but differ in the way of content presentation. Unlike the web, the contents are written by a single author or a discussion on a specific topic. Bloggers write whatever is on their mind, sometimes inventing new vocabulary and grammar. Some blogs deliberately deviate from rules of language and decorum to attract larger audience followers for their blogs [1]. Blogs are difficult to classify automatically as the author can write on any topic and present in his own style. Many of the blog directories are human edited and list the links after manual review. The exploding growth of heterogeneous contents on the web leads the automatic blog classification awfully difficult and huge time consuming. Several challenges are apparent in classifying the Blogs for a Blog directory, which include informal presentation of contents, type and structure of data, volume of the information and availability of millions of blogs etc.

The blogs can be classified according to their contents like political blogs, science blogs, fiction blogs etc. It can also be categorized by identifying the emotions of the author behind the article. The blog contents can be classified using document classification method and it can be recommended to users after user preference analysis. Since the readers may be interested in specific kind of articles, they have to filter the articles manually. For the effective information retrieval, a two-layer SVM classification mechanism to classify blog articles is proposed by Guo-HengLuo, Jia-Chiam Liu and Shyan-Ming Yuan [2]. Mita K. Dalal and Mukesh A. Zaveri[3] have classified unstructured blog posts using a semi-supervised machine learning approach. In their research they concluded that blogs can be classified with good accuracy using the multi-step classification strategy. The TF-IDF combined with Multi-word heuristics can be an effective statistical feature set extractor. They tested the classification of unstructured blog text using Naïve Bayes algorithm and basic artificial neural networks. The challenges in their research were larger and more varied datasets. Elisabeth Lexet al. [4] have classified the blogs into common newspaper categories using German News Corpus. The supervised text classification algorithms are generally applied to classify blogs into topics or other categories. The challenge in supervised text classifiers is it needs a sufficient large amount of labeled data to learn a good model. For blogs, data labeled with terms that capture current and actual topics are not available and data labeled in the past is not applicable owing to topic drifts. Their approach is to exploit the labeled data from the news corpus and use this knowledge to perform cross-domain classification on the unlabeled blogs. They evaluated their approach with a number of text classification algorithms with different parameter settings by means of accuracy and complexity. Their proposed Class-Feature-Centroid classifier (CFC) achieved a good accuracy.

In text categorization, the term weighting is used to discriminate the terms by assigning proper weights to the terms that result in better performance. The distinction between supervised and unsupervised methods is that the former use categorical information and the later are computed

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 9 (2018) pp. 6536-6544

corpus-wide [7]. Term Frequency (TF) and other variations of TF weighting schemes can be placed in the unsupervised category, including Term Frequency-Inverse Document Frequency (TF-IDF), Document Frequency (DF), Glasgow and Entropy. The supervised weighting techniques use the information about the class that the document belongs to in calculating the weight of the term. Man et al. [8] have investigated several widely used unsupervised and supervised term weighting methods in combination with SVM and KNN algorithms on benchmark data collections. They have proposed a supervised term weighting method based on relevance of documents in the corpus called TF-RF that improves the terms discriminating power for text categorization task. Feature selection is the basic phase in statistical-based text categorization and it depends on the term weighting methods. Thabit Sabbahet al. [7] have proposed four modified frequency-based term weighting schemes namely; mTF, mTFIDF, TFmIDF, and mTFmIDF to improve the performance of text categorization. Their proposed term weighting schemes consider the amount of missing terms into account calculating the weight of existing terms. ThabitSabbah et al. [6] have proposed hybridized term weighting method for identifying terrorism contents in web contents. The term weighting schemes TF, DF, TF-IDF, Glasgow and Entropy are used to compute term weight and two combination functions are used to reduce the feature set for effective classification.

Feature selection is a combinatorial optimization problem that selects the most important features from an original feature set [9]. It plays a pivotal role in categorizing the document effectively and efficiently. The traditional text categorization is based on term matching where a document is represented as the high dimensional vector space model. The rows in VSM represent the text documents and columns represent the words in the documents. The term matching categorization technique does not consider the semantic relationship between terms thereby result in a poor categorization. A two-stage feature selection method is proposed by Jiana Meng et al.[5] to categorize spam blogs, that reduce the dimension of terms and then build a new semantic space between terms, based on the latent semantic indexing method. Bing Xue et al. [10] have listed and stated in their study that the evolutionary computation (EC) paradigms can be used as search techniques in feature selection. It can be categorized as evolutionary algorithms, swarm intelligence and others. The evolutionary algorithms include genetic algorithm and genetic programming. Examples of swarm intelligence are particle swarm optimization and ant colony optimization. The EC paradigms include Differential evolution, memetic algorithms, evolutionary strategy and artificial bee colony etc.

Feature extraction [11] synthesizes a set of new set of features from the original features and it is smaller than the original feature set. Its outcome may not be a subset of the original feature. It can be result of combinations or transformations of the original feature space. Harun [12] in his methodology ranked each term in the document with respect to their importance in the document using Information Gain (IG). Then the Genetic Algorithm (GA) and Principal Component Analysis (PCA) are applied separately to the ranked terms and then the dimension is reduced based on their ranking. The k-nearest neighbor and C4.5 decision tree classification algorithm was used at the final stage to validate the dimension reduction methods. A novel clustering based feature subset selection framework was proposed by Sivakumar [13]. Initially clusters are formed using minimum variance method that is used to reduce the number of features. The cluster pair which has maximum number of votes is chosen and a member with the highest priority is chosen from each cluster using Information Gain (IG) and the attributes with less priority voting were removed resulted in dimensionality reduction.

The text classifier framework [14] usually involves three major phase, they are Term selection, Term weighting and Classifier learning. The most relevant terms are identified in term selection. Then the terms in the document are transformed using weighted function. Finally in the Classifier learning, a classifier is generated from the weighted representations of the training documents. This process is generally a supervised learning. Artificial neural network learning algorithms [15] are applied in many different applications such as classification, clustering, and pattern recognition by mimicking the behavior of the human neural system. The neural networks plays pivotal roles in categorization as, neural networks are data driven self-adaptive methods. The ANN can adjust the network parameters according to the data without any categorical specification of functional or distributional form for the underlying model.

In this paper classifying the Blogs of various categories with two phase feature extraction and pattern selection method using artificial neural network is proposed. The blogs are represented as feature sets. To determine the significance of each feature, they are weighted using term weighting method. The weighted terms are then ranked using Information Gain, which is a supervised feature selection method. Patterns with the salient features are given to the second phase for reducing the training patterns using leader algorithm. The article is organized as given. The Blog dataset representation and preprocessing techniques are discussed in Section 2. The term weighting schemes are listed in Section 3. The Information Gain for the feature extraction is explained in Section 4. The dataset reduction method using Leader algorithm is illustrated in Section 5. The Classifier with its training method is explained in Section 6. The Blog Classification framework and the proposed algorithm are defined in Section 7. The experiments and results are discussed in detail and compared with existing schemes in Section 8.

BLOG REPRESENTATIONS

The first step in the Blog classification is to transform a Blog page, which consists of Characters, Images, Hyperlinks and HTML Tags into a feature vectors. The preprocessing steps are carried out as shown in Figure 1.

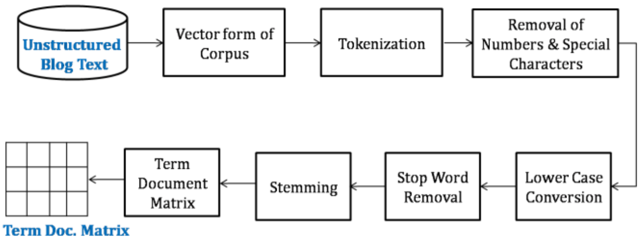

Figure 1: Blog Representation

A document can be represented as a document vector. The tokenization is a process of word segmentation. The words are chopped at white spaces in the sentences of the document and removes punctuations. Then the special characters and numbers are removed [17] as they don't play significant role in the blog content classification. All the words are uniformly converted into lower case since it is easy to compare them with the word in dictionary for stopping and stemming. Stop words are frequently occurred, most common words like 'the', 'in', 'and', 'whose', that carries less priority in categorization. There is no specific standard list of English stopwords, the common range of stop words are between 100 and 1,000 terms [18]. Most of the text analysis software packages make use of a default list for the stop word removal process. In addition to the default stop words list, the domain specific stop word dictionary can be constructed for yielding better performance. Stemming is a process that relates semantically similar indexing and search terms. Stemming is used to reduce the size of index terms [19]. Stemming can be done in different approaches that include affix removal, successor variety, table lookup and n-gram. The process is a vocabulary reduction technique intended to map a word to its most basic form. For example the words "sleep", "slept", "sleeping" all share a common stem "sleep". Consequently the large numbers of redundant and least important words are removed. Thus each Blog is represented by vector of 'm' unique words called as 'bag of words', d= {t1, t2, ... tm}. A term document matrix which is the vector space model of the dataset is constructed with each blogs as the rows, the index words or terms as column, and the cell values as the term frequency of the term in the respective document.

TERM WEIGHTING

The term value for a certain document specifies how much it influences the semantics of the document. The significance of a term cannot be measured by only with the frequency of occurrences. There are various schemes available to measure the weight of a term. They are listed in the Table 1 [20].

Table 1: List of Term Weighting Methods

Term Weighting Method	**Formula**
Term Frequency (TF)	$\mathrm{TF}_{t,d} = \dfrac{\mathrm{fr}_{t,d}}{\sqrt{\sum_{t=1}^{n} \mathrm{fr}_{t,d}{}^{2}}}$
Document Frequency (DF)	$\mathrm{DF}_t = \sum_{d=1}^{N} \begin{Bmatrix} 1 & t \in d \\ 0 & t \notin d \end{Bmatrix}$
Term Frequency – Inverse Document Frequency (TF-IDF)	$\mathrm{TF-IDF}_{t,d} = \mathrm{TF}_{t,d} \,.\, \mathrm{IDF}_t$ $\mathrm{where IDF}_t = \log\left(\dfrac{N}{\mathrm{DF}_t}\right) + 1$
Entropy	$w_{t,d} = L_{t,d} \mathrm{x}\, G_t$ $\mathrm{where}G_t = \dfrac{1 + \sum_{j=1}^{N} \frac{\mathrm{fr}_{t,d}}{F_t} \log(\frac{\mathrm{fr}_{t,d}}{F_t} + 1)}{\log N}$ $\mathrm{and}\ L_{t,d} = \begin{Bmatrix} 1 + \log \mathrm{fr}_{t,d} & , & \mathrm{fr}_{t,d} > 0 \\ 0 & , & \mathrm{fr}_{t,d} = 0 \end{Bmatrix}$
Term Weighting[20]	$\boldsymbol{w_{t,d} = \dfrac{\log(TF_{t,d} + 1)}{\log(\frac{DF_t}{N} + 1)}}$

The symbols in Table 1 denotes,

$\mathrm{fr}_{t,d}$ is the frequency of term 't' in document 'd'

F_t is the frequency of term't' at the document collection level

n is the number of unique terms in document 'd'

N is the number of documents in the collection

length_d is the length of the vector that represents unique terms in document 'd'

i. Term Frequency (TF): TF is the direct method that states how many times the term ‘t’ occurred in the document ‘d’. In TF scheme, if a term frequency is high it shows that term is more relevant than the term with low frequency.

ii. Document Frequency (DF): Document frequency enumerates how many documents in the document collection has the term ‘t’.

iii. Term Frequency – Inverse Document Frequency (TF-IDF): The Inverse document frequency implies the most occurred terms in the collection are least significant terms. TF-IDF is a prominent global term weighting scheme where the weight is computed with respect to its incidence in the entire collection. In this inevitable ranking measure the term which is less frequent in the collection however most occurred in a specific document is assigned higher weight.

iv. Entropy term weighting: The term weight is computed from two aspects. They are local term weighting and global term weighting based on purity measure.

v. Term weighting [20]: This term weighting method is enhanced from traditional TF-IDF. It majorly focus on three aspects, they are term frequency, collection frequency and document length.

INFORMATION GAIN

Information Gain Information Gain [12] is a supervised technique based on Information theory used for ranking the attributes. The Information Gain Ratio is initially introduced to measure the goodness for attributes used in Decision Tree learning algorithm. It measures the no. of bits of information acquired for estimation of a class (C) by knowing the availability of a term (t) in a document. The information gain of term ‘t’ is defined as,

$$\mathrm{IG}(t) = -\sum_{i=1}^{|C|} P(C_i)\log P(C_i) + P(t)\sum_{i=1}^{|C|} P(C_i|t)\log P(C_i|t) + P(\bar{t})\sum_{i=1}^{|C|} P(C_i|\bar{t})\log P(C_i|\bar{t}) \quad (1)$$

Where Ci represents the ith Class and P(Ci) is the Probability of the ith Class. P(t) and P($\bar{t}$) are the probabilities that the term ‘t’ is present or not present in the documents respectively. P(Ci|t) and P(Ci|$\bar{t}$) are the conditional probabilities of the ‘ith’ class given that the term ‘t’ does not appear.

LEADER ALGORITHM

Leader algorithm [21] is an unsupervised clustering technique that groups the data sets according to the similarity. The similarity is computed by standard distance measure. Clustering is formulated as an optimization problem to create a subset of clusters. If D= {d1, d2, d3...dn} is the collection of documents; ‘n’ is the maximum no. of documents in the dataset; ‘k’ is the maximum no. of clusters formed by Leader Cluster algorithm and C= {c1, c2, …ck} are the centroids of the clusters. Leader algorithm is an incremental clustering algorithm used to cluster large data sets. This algorithm is order dependent and may form different clusters according to the input order, the data set is provided to the algorithm. The algorithm consists of the following steps.

Algorithm: Leader Cluster

Input: A Collection of documents ‘D’

Output: New Subset of clusters ‘K’

1. Read the first data item, d_1 and allocate it to the first cluster C_1. This data is the leader of the cluster C_1.
2. Increment no. of clusters to 1
3. Read the next data item d_2 and calculate its distance from the leader d_1.

   The distance is computed using Euclidean distance measure as in (2)

   $$dist = \sqrt{(d_1 - d_2)^2} \quad (2)$$
4. If the distance between d_2 and leader d_1< threshold t, then data point d_2 is assigned to cluster C_1.
5. If the distance between d_2 and leader d_1>threshold t, then form a new cluster C_2 and assign d_2 to this new cluster and d_2 will be the leader of the cluster C_2.
6. Repeat the steps 6 – 10 for all the remaining data items.
7. Calculate the distance between the data point and the leader of the all the clusters
8. If the distance between the data items and the any of the leader < threshold t, the data point is assigned to that cluster.
9. If the computed distance for all the clusters is greater than the threshold, a new cluster is created and the data point is assigned to that cluster. Now this data is the leader of the new cluster.
10. Increment no. of clusters to 1

CLASSIFIER

Artificial Neural Network (ANN) is prominently used as a classifier on implementing two major steps. First, Construction of architecture of ANN secondly, training the ANN. The network can be constructed by connecting neurons or the functional elements in different layers. The most frequently used topology is feedforward neural network which can be constructed by having minimum of three layers as shown in Figure 2. The multiple layers include input layer, output layer and hidden layer. The neurons in the input layer are determined by the number of input features and similarly number of output neurons is determined by the no. of outputs. The hidden neuron can be adjusted according to the application and the training. All the neurons are connected in a standard manner using weighted links. The training of ANN is a classification problem where the weights of the links are adjusted in order to receive better accurate output. Back-propagation training algorithm is used for training the network. With the algorithm the network is continuously observed and learnt by gradually reaching to the specified Mean Square Error (MSE) value. The MSE is computed as

$$MSE = \frac{1}{p}\sum_{i=1}^{p}(d_i - o_i)^2 \qquad (3)$$

where 'p' denote the total number of patterns, 'j' denote the jth neuron of the output layer, 'd' is the desired value and 'o' is the computed output of the network for the given pattern. The basic idea of the standard backpropagation algorithm is the recurrent use of the chain rule to calculate the impact of each weight in the network with respect to an arbitrary error function.

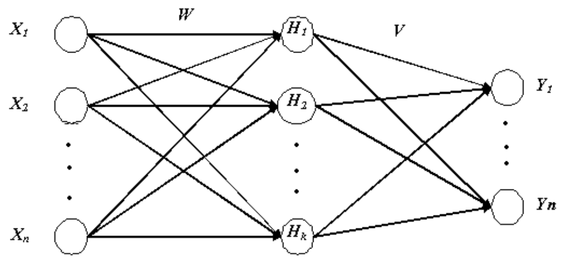

Figure 2: Feedforward Neural Network

PROPOSED METHOD

A blog has contents more about personal opinions, activities, and experience. The massive contents of the blog lead to dimensionality problem and will have the adverse effects on the performance of the classifier. The blogs can be classified according to the contents; emotions of the blogger; professionalism etc. In this paper the blogs are classified according to their category as shown in the Figure 3. The blogs which are retrieved from WWW has contents in the form of text, video, images etc., are arranged in chronological order. For the classification of blogs according to category, first, contents alone are extracted by removing all html tags and links. Then in the preprocessing phase, the text is tokenized and then stop words are removed. Then Stemming is also applied to map the modulated or derived words to their word stem which reduces the term index to a considerable amount. The result of the preprocessing phase is a collection of unique term index and they are formulated as the document-term frequency matrix as in Table 2.

Table 2: Sample DTM

Blog \ Term	t_1	t_2	t_3	…	t_k
$Blog_1$	21	7	10	…	8
$Blog_2$	17	11	5	…	9
$Blog_3$	8	6	3	…	4
:	:	:	:	…	:
$Blog_m$	2	14	5	…	15

The rows and columns of the table represent the documents and the terms respectively. In the table, m denotes the total number of blog documents and k is the total number of terms present in the document collection. The values in the cells of the table depict the number of times the term t_i available in $Blog_j$.

The Vector Space Model (VSM) of the given data has too many terms out of those very few terms are significantly relevant. Thus, the terms are weighted using the term weighting formula and weights are sorted in an ascending order. There can be specific number of features given to the next level of feature ranking, which is done using Information gain, a supervised technique. Now top ranked *p* numbers of features are selected for the next phase. The selected features undergo normalization so that the data sets are scaled within the range of [1, −1]. Later the feature reduction, the pattern reduction is done using Leader cluster algorithm. The algorithm significantly reduces the patterns and forms 'k' clusters depend on the threshold distance. The leader patterns are given as inputs to train ANN classifier. The feedforward artificial neural network categorizes the given dataset.

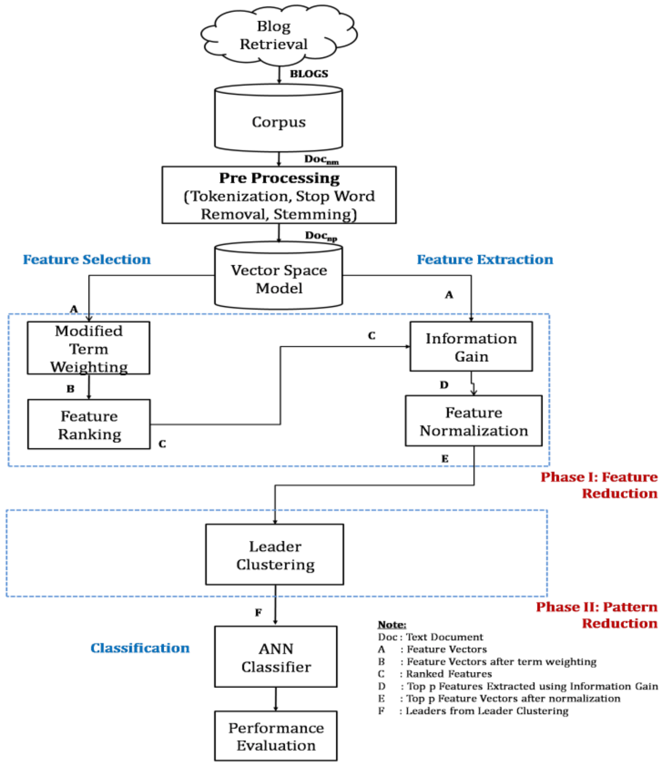

Figure 3: Blog Classification

BLOG CLASSIFICATION ALGORITHM

Step 0: Start

Step 1: Consider p no. of features; k no. of words in the collection; m no. of blog documents in the collection

Step 2: Retrieve Blogs; Extract only blog text contents

Step 3: Perform the text preprocessing techniques

Step 4: Construct Vector Space Model for the preprocessed text

Step 5: Input the VSM of the blog documents in matrix A (Dimension m x k)

Step 6: Transform the matrix A to Term weight matrix B using the improved term weighting function [20]

Step 7: Sort the matrix B in descending order and rank the columns with respect to the sum of term weight

Step 8: From the sorted matrix B, extract the top ranked 5000 features to matrix C.

Step 9: Perform supervised ranking using information gain

Step 10: Extract the top ranked p no. of features

Step 11: Normalize the feature reduced matrix with dimension m x p

Step 12: The normalized output E is given as input to Leader algorithm

Step 13: Perform Leader clustering that reduces the rows

Step 14: Feed Leaders of the clusters to ANN

Step 15: Initialize the network parameters for FNN

Step 16: Train the ANN, until the MSE reaches 0.001

Step 17: Feed the testing documents as input to the network

Step 18: Compute the classifier performance by measuring Accuracy, Precision, Recall and F1

EXPERIMENTS AND RESULTS

Experiments have been conducted on the dataset contains blog posts from MarginalRevolution.com [16]. It has the collection of posts from Jan. 1, 2010 to 9/17/2016, with the attributes as Author Name, Post Title, Post Date, Post content (words), Number of Words in post, Number of Comments in post, and categories. This dataset has about 13,000 blogs, subdivided into 15 classes. 1612 documents (m=1612) of 5 classes are used for the analysis of the proposed method. Table 3 shows the details of various blog posts taken from the dataset for the experiment. The number of features are limited to 5000 (k=5000). The data is cleaned by applying text preprocessing techniques using R language. The Hardware and Software specifications for the execution of the experiment are shown in Table 4.

Table 3: Marginal revolution Blogbost

Category	***No. of Documents***
Economics	397
Web Technology	468
Books	349
Food & Drinks	225
Political Science	173
Total	1612

Table 4: Hardware and Software Configuration

Hardware	**Software**
Processor: Intel Core Duo 2.1GHz	Platform: MS Windows 7
Memory: 3GB RAM; 32 bit OS	Software: Matlab R2014a, R Studio

At the end of the preprocessing phase there are ‘m’ documents and ‘n’ terms in the document term matrix. The terms are weighted using term weighting techniques and sorted in the descending order of the term values. The top ranked 5000 columns are fetched for the Information gain ranking. The weighted matrix is ranked with supervised ranking and ‘p’ no. of significant features are selected. The outcome of the Feature selection and extraction is a matrix of ‘m’ documents and ‘p’ columns. At this juncture, it is necessary to assign the optimal value for ‘p’ in the first phase and ‘distance’ for leader cluster in the next phase of the proposed framework. The parameter ‘p’ is given with different values (p=25, 30, 35 and 40) in accordance with leader distance (dist= 1.0, 1.5, 2.0, 2.5 and 3.0) and the accuracy results are tabulated in Table 5. It is found that better results are yield for p=25 and dist=2.0 results in compact and efficient dimensionality reduction. The p is chosen as 25 and the resultant matrix with reduced columns undergoes an unsupervised clustering, Leader clustering for the row reduction. The distance for leader clustering is chosen as 2.0. The leaders of the clusters are extracted for the classification. The feedforward neural network is configured with 25 neurons in input layer, 8 neurons in hidden layer and 5 neurons in the output layer. The network is trained using backpropagation algorithm. The parameters of the feedforward ANN are given in Table 6. On trial basis, the learning rate is fixed for minimum cost. Initially the weights are chosen randomly between 0 and 1. The learning curve of the configured network is shown in Figure 4.

Table 5: No. of features, leader distance Vs Accuracy

p	**Leader distance**	**No. of Leaders**	**Accuracy (%)**
25	1.0	78	55.1
	1.5	34	73.5
	2.0	**18**	**92.9**
	2.5	2	-
	3.0	1	-
30	1.0	99	51.9
	1.5	45	68.9
	2.0	23	89.5
	2.5	3	-
	3.0	1	-
35	1.0	122	53.3
	1.5	50	62.0
	2.0	28	85.0
	2.5	5	-

p	Leader distance	No. of Leaders	Accuracy (%)
	3.0	2	-
40	1.0	146	53.4
	1.5	58	60.3
	2.0	33	82.6
	2.5	6	-
	3.0	2	-

Table 6: ANN parameters

Parameter	Value
Learning Rate	0.1
Mean Squared Error	0.001
Input Neurons	p [25]
Hidden Neurons	8
Output Neurons	5

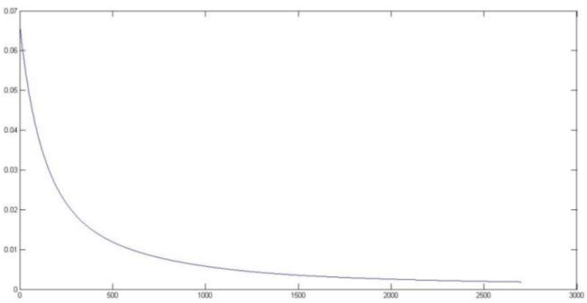

Figure 4: Epoch Vs MSE Learning Curve

Once the classifier is trained, it is tested with set of patterns and the results are evaluated using standard information retrieval measurement tools that are precision (P), recall (R), Accuracy (Acc) and F1.

$$P = \frac{a}{a+b} \quad (4)$$

$$R = \frac{a}{a+c} \quad (5)$$

$$F1 = \frac{2PR}{P+R} \quad (6)$$

$$Acc = \left(\frac{Total\ Correct}{Total\ No.of\ Documents}\right) x100\% \quad (7)$$

The Figure 5 illustrate the confusion matrix of blog classification using proposed method with the p value as 25 and cluster distance as 2.0. It shows that out of 14 patterns 13 patterns are predicted correctly. In the five classes of patterns, only Economics class dataset is predicted wrongly. A pattern in the Economics dataset is predicted as Food and drinks class, due to similarity in the keywords. The confusion matrix for blog classification using proposed method and TF-IDF is shown for web technology class in Figures 6 and 7.

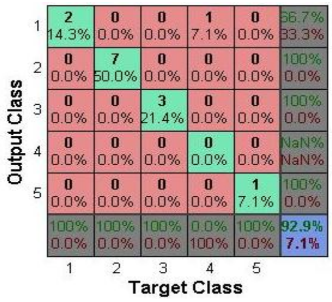

Figure 5: Confusion Matrix for Blog Classification using proposed weighting scheme

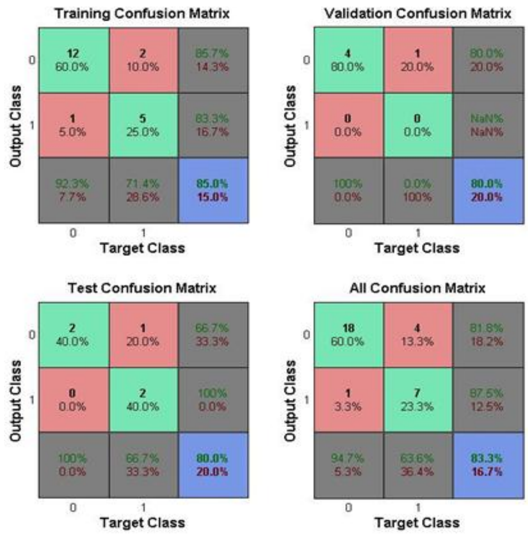

Figure 6: Confusion Matrix for Web technology Blog Classification using proposed weighting scheme

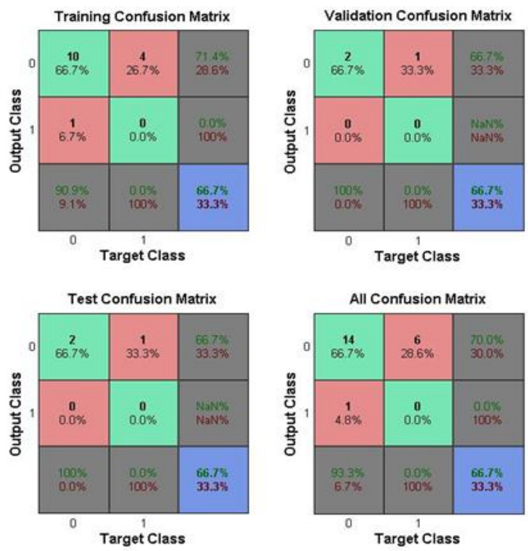

Figure 7: Confusion Matrix for Web technology Blog Classification using TFIDF weighting scheme

The Figure 6 shows the confusion matrix of web technology class using the proposed method, where 25 patterns are predicted correctly and 5 patterns are predicted wrongly out of 30 patterns. The 4 patterns were false positive and 1 was false negative. Figure 7 shows the misclassification results of TFIDF scheme for web technology class. 7 patterns are predicted wrongly and 14 patterns are predicted correctly.

Table 7: Comparison of Classification Accuracy

Class	TF	DF	TF-IDF	Entropy	Proposed Method
Economics	90.4%	86.1%	85.7%	94.6%	**96.7**%
Web Technology	68.5%	64.3%	66.7%	74.6%	**83.3**%
Books	85.4%	81.2%	87.7%	90.3%	95.2%
Food & Drinks	80.4%	77.7%	85.6%	88.6%	90.4%
Political Science	82.6%	81.1%	83.5%	83.75%	85.9%

Table 8: Overall performance of the proposed method

Class	Accuracy	Precision	Recall	F1
Economics	**96.7**%	100%	94.7%	97.2%
WebTechnology	**83.3**%	81.8%	72.0%	76.2%
Books	95.2%	95.8%	100%	97.9%
Food & Drinks	90.4%	100%	94.4%	97.2%
Political Science	85.9%	93.3%	87.5%	90.3%

Similarly for the remaining four classes the experiments were conducted and tabulated in Table 7. The patterns predicted under false positive are more than false negative in both the methods. It is observed that even though the patterns are reduced more compactly using TFIDF the accuracy is less compared to the proposed scheme. The classification accuracy of the proposed method by varying different term weighting methods is tabulated in Table 7. The accuracy of the proposed scheme is found to be remarkable on comparing all the weighting schemes. The classifier performance in terms of accuracy, precision, recall and F-measure is tabulated in Table 8.

CONCLUSION

Blog classification is the web matching process that allocates a blog to one of the predefined category. It can help search engines to effectively extract the data and rank them. The incredible growth of web leads to the curse of dimensionality during the classification. In this paper a two phase dimensionality reduction and classification is implemented. The proposed hybrid method uses both supervised and unsupervised techniques for dimensionality reduction. Since the significant features and patterns after reduction is given as input to the classifier, the results shows better accuracy. The experiments were conducted on blog posts of five classes and the results illustrate that the average results are improved than the existing methods. The work can be extended for the multiple class blog posts and with different classifiers.

REFERENCES

[1] Hong, Qu., Andrea, LA., Pietra and Sarah Poon., 2006, “Automated Blog Classification: Challenges and Pitfalls”, Proc. AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs 2006: pp. 184 – 186.

[2] Guo-Heng, Luo., Jia-chiam, Liu., and Shyan-Ming Yuan., 2011, “A Two-Layer SVM Classification Mechanism for Chinese Blog Article”, Proc. International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Oct 2011, pp. 9 -15.

[3] Mita K. Dalal., Mukesh A. Zaveri., 2013, “Automatic Classification of Unstructured Blog Text”, Journal of Intelligent Learning Systems and Applications. **5**, pp. 108-114.

[4] Elisabeth Lex., Christin Seifert., Michael Granitzer., and Andreas Juffinger., 2009, “Automated Blog Classification: A Cross Domain Approach, Proc. International Association for Development of the Information Society, International Conference on WWW/Internet, 2009, pp. 598 - 606.

[5] Jiana Meng., Hongfei Lin., and Yuhai Yub., 2011, “A Two-stage Feature Selection Method for Text Categorization”, Computers and Mathematics with Applications, 62, pp. 2793–2800.

[6] Thabit Sabbah., Ali Selamat., Md Hafiz Selamat., Roliana Ibrahim., and Hamido Fujita., 2016, “Hybridized Term Weighting Method for Web Contents Classification using SVM”, Neuro Computing, 173, pp. 1908 -1926.

[7] Thabit Sabbah., Ali Selamat., Md Hafiz Selamatb., Fawaz S. Al-Anzid., Enrique Herrera Viedma., Ondrej Krejcar., and Hamido Fujita., 2017, “Modified Frequency-basedt Term Weighting Schemes for Text Classification”, Applied Soft Computing, 58, pp. 193 – 206.

[8] Man Lan., Chew Lim Tan., Jian Su., and Yue Lu., 2007, “Supervised and Traditional Term Weighting Methods for Automatic Text Categorization”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 10, pp. 721 - 735.

[9] Sina Tabakhi., Parham Moradi., Fardin Akhlaghian., 2014, “An Unsupervised Feature Selection Algorithm based on Ant Colony Optimization”, Engineering Applications of Artificial Intelligence, 32, pp. 112 - 123.

[10] Bing Xue., Mengjie Zhang., Will N. Browne., and Xin Yao., 2016, "A Survey on Evolutionary Computation Approaches to Feature Selection", IEEE Transactions on Evolutionary Computations, 20(4), pp. 606 – 620

[11] Choi, B., and Yao, Z., 2005, "Web page classification", In W. Chu & T. Lin (Eds.), Foundations and Advances in Data Mining, pp. 221-274.

[12] Harun Uguz., 2011, "A Two Stage Feature SelectionMethod for Text Categorization by using Information Gain, Principal Component Analysis and Genetic Algorithm", Knowledge Based Systems, 24, pp. 1024 - 1032.

[13] Sivakummar Venkataraman., Subitha Sivakumar., and Rajalakshmi Selvaraj., 2016, "A Novel Clustering based Feature Subset Selection Framework for Effective Data Collection", Indian Journal of Science and Technology, 9(4), pp. 1-7.

[14] Franca Debole., and Fabrizio Sebastiani., 2003, "Supervised Term Weighting for Automated Text Categorization", Proc. 18th ACM Symposium on Applied Computing (SAC 2003), pp. 784-788.

[15] Jayanta Kumar Basu., Debnath Bhattacharyya., Taihoon Kim., 2010, "Use of Artificial Neural Network in Pattern Recognition", International Journal of Software Engineering and Its Applications, 4, pp. 23- 33.

[16] https://www.kaggle.com/wpncrh/marginal-revolution-blog-post-data

[17] Matthew J Denny., and Arthur Spirling., 2017, "Text Preprocessing for Unsupervised Learning: Why It Matters, When It Misleads, and What to Do about It", (September 27, 2017), SSRN: https://ssrn.com/abstract=2849145 or http://dx.doi.org/10.2139/ssrn.2849145

[18] http://www.ranks.nl/stopwords

[19] W. B. Frakes., 1992, "Stemming Algorithms", In W. B. Frakes and R. Baeza-Yates, Editors, Information Retrieval: Data Structures & Algorithms, Prentice Hall, Englewood Cliffs, NJ, 1992.

[20] Kathirvalavakumar Thangairulappan., and Aruna Devi, Kanagavel., 2016, Improved Term Weighting Technique for Automatic Web Page Classification, Journal of Intelligent Learning Systems and Applications, 8, pp. 63 - 76.

[21] P.A Vijaya., MNarasimha Murty., and D. K Subramanian., 2004, "Leaders–Subleaders: An efficient hierarchical clustering algorithm for large data sets", Pattern Recognition Letters, 25(4), pp. 505 - 513.

[22] Abualigah L.M., Khader A.T., Hanandeh E.S., 2018, "A Novel Weighting Scheme Applied to Improve the Text Document Clustering Techniques", In: Zelinka I., Vasant P., Duy V., Dao T. (eds) Innovative Computing, Optimization and Its Applications, Studies in Computational Intelligence, Vol 741. Springer, Cham, pp. 305 – 320.