Web page Classification by Using PCA and Neural Network

Laith R. Flaih

Computer Science Department, College of Science, Cihan University-Erbil

Email: d.lath1974@yahoo.com

Abstract

With the exclusive growth in the WWW makes the internet growing very fast. Therefore classifiers of the web pages become more challenging. The proposed system is about using Principal Components Algorithm PCA to classify web documents . In this research, new web page classification method is proposed, and the proposed system uses a neural network with inputs obtained by the Principal Components Algorithm. The feature vectors that obtained from PCA are then used as the input to the neural networks for classification. The experimental evaluation demonstrates that the proposed system provides high quality classification accuracy with the sports news datasets.

Keywords: PCA ,Neural Networks, Web pages Classification, classifiers

1. Introduction

Classification is one of the most frequently encountered decision making tasks of human activity. A classification problem occurs when an object needs to be assigned into a predefined group or class based on a number of observed attributes related to that object.

Neural networks have emerged as an important tool for classification. The recent vast research activities in neural classification have established that neural networks are a promising alternative to various conventional classification methods (Zhang, 2000).

Here, the proposed system is a web page classification method, which base on the PCA. Each web page is represented by the term frequency-weighting scheme.

In order to classify the news web pages, the proposed system uses the PCA as the input to the neural networks. We have used the PCA algorithm to reduce the original data vectors to a small number of relevant features.

DOI: 10.24086/cuesj.v1n1a12

The PCA can be computed by an Eigen value decomposition of the covariance matrix of the input process (Weingessel and Hornik, 2000).

PCA is the simplest of the eigenvector-based multivariate analyses. Often, its operation can be thought of as revealing the internal structure of the data in a way which best explains the variance in the data (Minaei-Bidgoliand and Punch III, 2003).

The organization of this paper is as follows: the Classifier is described in section 2. The Neural Classifier explained in section 3. After this preprocessing and the overview of the propose system which describe the classification process explained in section 4. And the last section gives the conclusion of this paper

.

2. Classifiers

Pattern recognition has a wide variety of applications in many different fields, such that it is not possible to come up with a single classifier that can give good results in all the cases. The optimal classifier in every case is highly dependent on the problem domain. In practice, one might come across a case where no single classifier can classify with an acceptable level of accuracy. In such cases it would be better to pool the results of different classifiers to achieve the optimal accuracy. Every classifier operates well on different aspects of the training or test feature vector. As a result, assuming appropriate conditions, combining multiple classifiers may improve classification performance when compared with any single classifier (Miltsakaki and Troutt, 2009 and Wikipedia, 2013).

Classifier systems should not be confused with noun classes, which often categorize nouns in ways independent from meaning, such as according to morphology (Ipeirotis, 2004).

We expect that the accuracy of the classifier will improve but we also expect that for very fine thematic distinctions alternative approaches may be required (e.g., give special weights for key vocabulary that will distinguish between sports subthemes) or develop new classification features beyond statistical analysis of word distributions (Blog, 2013).

Classifier quality can be increased with more training data, but creating large numbers of training examples might be prohibitively expensive (Chakrabarti, 2004)

Basically what a classifier does is assign a pre-defined class label to a sample. For example, if you are building a spam classifier then the feature space contains a representation of an email and the label is either “Spam” or “Non-Spam” (Scime, 2005).

The classifier analyzes correlations between the labels and other document attributes to form models. Later, the classifier is presented with unlabeled instances and is required to estimate their topics reliably (Jo, 2008).

Naïve Bayesian classifiers (Friedman & Kohavi, 2002; Mitchell, 1997; Weiss & Kulikowski, 1991) are a popular technique used for classification and especially popular for the classification of text (Mora-Jime´nez and Figueiras-Vidal, 2009).

3. Neural Network Classifiers

Since the proposed neural network is intended originally only for text categorization, it is called NTC (Neural Text Categorizer).

The proposed neural network follows Perceptron in that synaptic weights are connected directly between the input layer and the output layer, and the weights are updated only when each training example is misclassified (BMVA, 2013).

Conventional search procedures for training neural classifiers equally use all the examples to minimize a sample estimate of the selected cost function; however, the real problem is to define appropriate classification borders, which is not exactly equivalent to any of these procedures, but that is the key to obtain good generalization. Thus, the possibility of having some examples more relevant for a good training (for a good definition of borders) must be accepted, and these examples have to receive a higher weight in the cost function estimate to be minimized. The subsequent problems are to determine what are these samples and how to emphasize them, a process which is named sample selection or, better, sample editing. Consequently, to design effective sample editing methods is a key subject in order to construct high performance neural classifiers (Fahmi, 2004).

The performance of neural network techniques themselves is always somewhat disappointing in relation with their computational effort and compared with more dedicated techniques and with its eternal competitor, the nearest neighbor rule (Resample, 2013).

The neural network classifier has become one of the main approaches in the automated text classification. It generates a classifier from the training set based on the characteristics of the documents already classified. Then it uses the classifier to classify the new documents (Fahmi, 2004b)

In the training phase, the correct class for each record is known (this is termed supervised training), and the output nodes can therefore be assigned "correct" values --

"1" for the node corresponding to the correct class, and "0" for the others. (In practice it has been found better to use values of 0.9 and 0.1, respectively.) It is thus possible to compare the network's calculated values for the output nodes to these "correct" values, and calculate an error term for each node (the "Delta" rule). These error terms are then used to adjust the weights in the hidden layers so that, hopefully, the next time around the output values will be closer to the "correct" values (Resample, 2013b).

4. The Proposed System

This section gives an overview data collection phase, preprocessing phase, term indexing constructing the classifier, neural class prediction.

Data collection phase process consists of crawling web site and getting that particular document from site and then cleansing the document and storing it into database.

In preprocessing phase it takes the HTML document and converts the stream of characters into stream of words after that string tokenizer process is applied in which every word is taken as token from this stop words elimination is done which means the prepositions like to, the, and etc. are removed from the document. Then stemming process is applied in which group of words is reduced to their grammatical roots.

Term Indexing is the process in which we are going to extract the HTML features and rank those features.

Constructing the classifier in this research will be using algorithms PCA, PCA is used for feature reduction.

In the following part we are going to explain about each and every phase and it sub parts in the proposed system and how it works.

4.1 Data collection phase

In this section we explain the two operations used for data collection phase and they are web crawling and document cleansing.

4.1.1 Web crawling

This process in mainly used to crawl the web sites and get the HTML document from that web site. This process has different names like bot, indexers, and spiders. We can use this crawling to get up to date data as well as we can maintain this document for

later processing by storing into the database. By using this kind of process we can make our information retrieval very efficient as shown in the Figure 1.

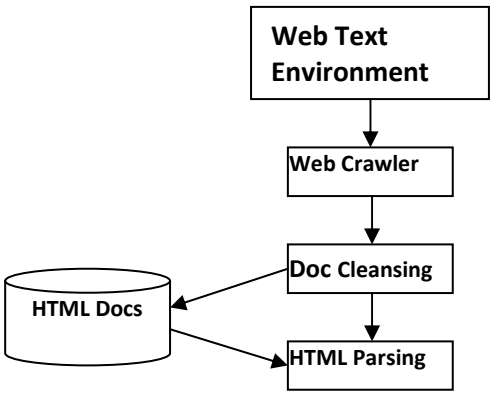

Fig. 1: The process of collecting data from the web

4.1.2 Document cleansing

This is the process of cleaning non-HTML files and words which are not meaningful, words mixed with special symbols and numbers. By maintaining HTML files in the document it makes search engine makes easier and increase the accessibility. After the completion of these two parts the document will be stored in database for further processing.

4.2 Preprocessing Phase

In this section we explain all operations used in preprocessing phase like lexical analysis, string tokenize, stop word elimination and stemming.

4.2.1 Lexical analysis

It is the process of converting the stream of characters into a stream of words the objective of lexical text analysis is to identify the words in text document. Where the stream of characters is given as input to the complier it scans that from left to right and produces a stream of words.

4.2.2 String Tokenize

Here each word is represented as token. After words as tokens it is easy to identify words in that plain text. Preprocessing technique is implemented to decrease the storage space of document.

4.2.3 Stop Word Elimination

In this we are going to define a group of stop words in a list such as (the, which, it, at, in, and etc), then we remove those words which can decrease the space of the document to store it into the database. If we take prepositions like to, the, and, into etc. these words are also of no use in classification process. So these words have been removed to reduce the number of words and which makes classification process easy, notify that we define every words that we think is not useful possible in the classification.

4.2.4 Stemming

Stemming is technique which used to reduce the words to their grammatical roots. This kind of technique is useful in the form of retrieving data or information and applications of data mining. For example take words like comparing, compare, compared, compares I have rooted the word to compare so that it will be useful during retrieving information. The stemming process is applied to remove suffixes like ed, ing, ily etc. by removing this suffixes we can reduce the terms in the document. This not only reduces the size of document but reduces the complexity also it is necessary to do text analysis to make information retrieval efficient in especially in data mining applications. This is called simple stemming process. From this will have two vectors as output one is abnormal vector and no stemmed words. Where abnormal vector has the words which does not have correct meaning and normal vector has the words with correct meaning. Below the two techniques are explained how it works:

The first technique is used to extract the words from document these extracted words might not be meaningful words after stemming i.e. not English words that is remained in the document.

The second technique is used to check the words in the document it has dictionary to check the stemmed words if the stemmed word is equal to the word in dictionary then it remained as it is because it is a meaningful word else the word cannot be stemmed. This not only increases the accuracy but also the access time.

See Figure 2 which shows the process of preprocessing of the html documents.

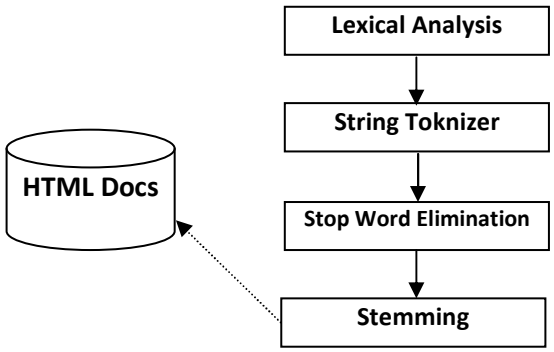

Fig. 2: The Process of Preprocessing of the HTML Docs

4.3 Term indexing phase

In this section we explain the two operations used in term indexing phase like feature extraction and HTML ranking.

4.3.1 Feature extraction

In this phase we take the particular HTML document and extract the features of that document. Features are nothing but HTML tags like <html>, <h1>, <keywords>, <a>, etc., here first thing is the extraction of the most important tags that can be help us in the classification for example the tag of title mean that text which occurs between this tag is the title of the subject then it is important to classification process to base on such tags, this process is done before classification.

4.3.2 HTML ranking

After extracting the selected tags the next step is going to see how many times the particular tag has been repeated then that needs to calculate the count of that tag. Here the count is nothing but ranking suppose if <html> tags has repeated for 5 times then the tag here is <html> and the rank of <html> tag is 5.

4.4 Classifier Construction Phase

In this part the web page classification methods is proposed that can be used the Principal Component Analysis (PCA) as the input to the neural networks, firstly that

must be to use the PCA for reducing the features and then regular words exist in each class as shown in the Figure 3

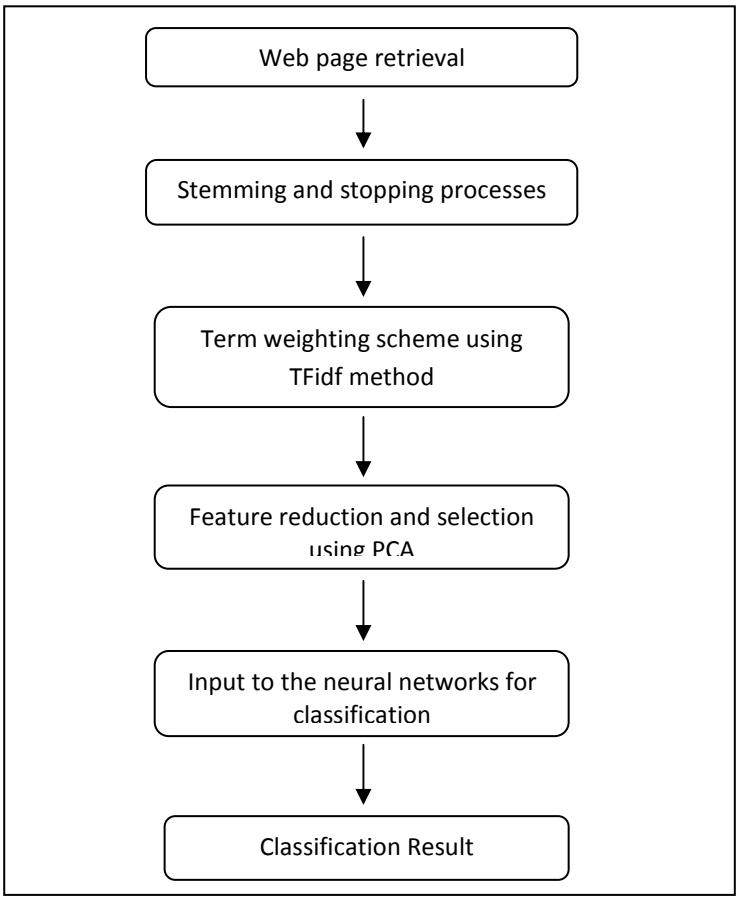

Fig. 3: The classification process of web page

4.4.1 PCA

As pointed in the previous section the PCA shorts for Principal Component Analysis that can be used it for the feature reduction, the main idea in the PCA is to calculate the mean and covariance of the matrix T, where T is a matrix document terms weight, then determining the Eigen values and eigenvectors of the covariance matrix C which is a real symmetric positive matrix. Suppose that C is the matrix of the covariance then the Eigen value and Eigen vector $Ce=\lambda e$ can be found where e is an Eigen vector of C. In order to find a nonzero vector e the characteristic equation:

$$|C - \lambda I| = 0 \quad \text{.....} \quad \text{....(1)}$$

Must be solved. If S is an m×m matrix of full rank, m Eigen values λ1, λ2,…..,λm can be found. By using

$$(S - \lambda I)e = 0 \quad ... \qquad .....(2),$$

all corresponding eigenvectors can be found. The Eigen values and corresponding eigenvectors will be sorted so that λ1≤λ2≤….≤λm.

In order to get the principal components of matrix C, that will perform Eigen value decomposition that is given by C=eΛeT. Then we select the first d ≤ m Eigen vectors where d is the desired value, e.g., 100, 200, 400, 600 etc. The set of principal components is represented as

$$Y_1 = e_1^T x,\ Y_2 = e_2^T x,...,\ Y_d = e_d^T x \quad ..... \qquad .....(3)$$

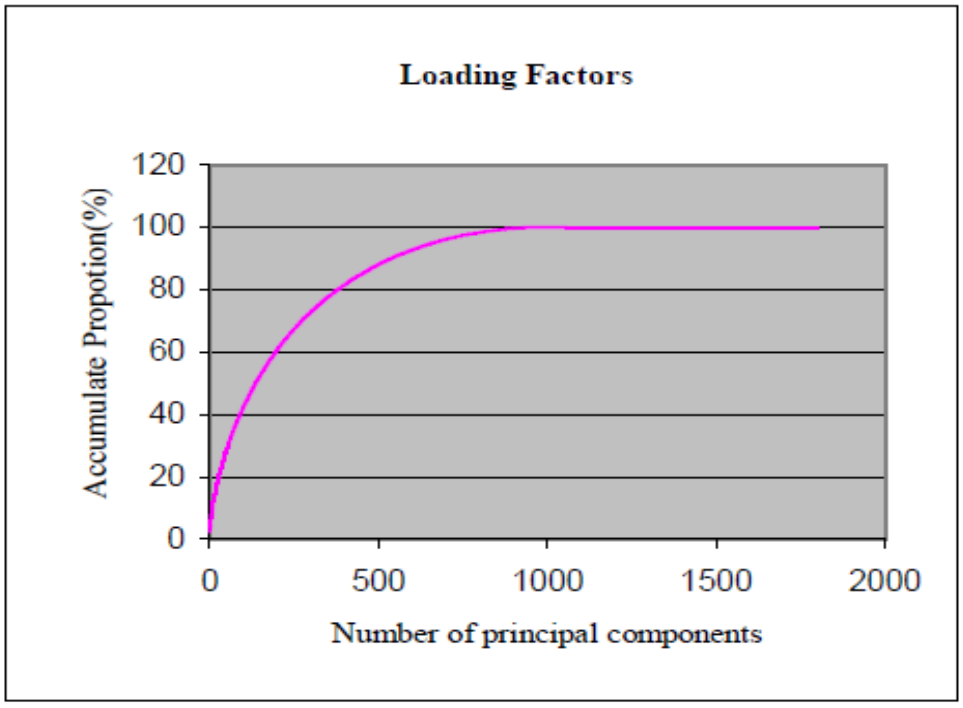

Fig. 4: Accumulated proportion of principal components generated by PCA

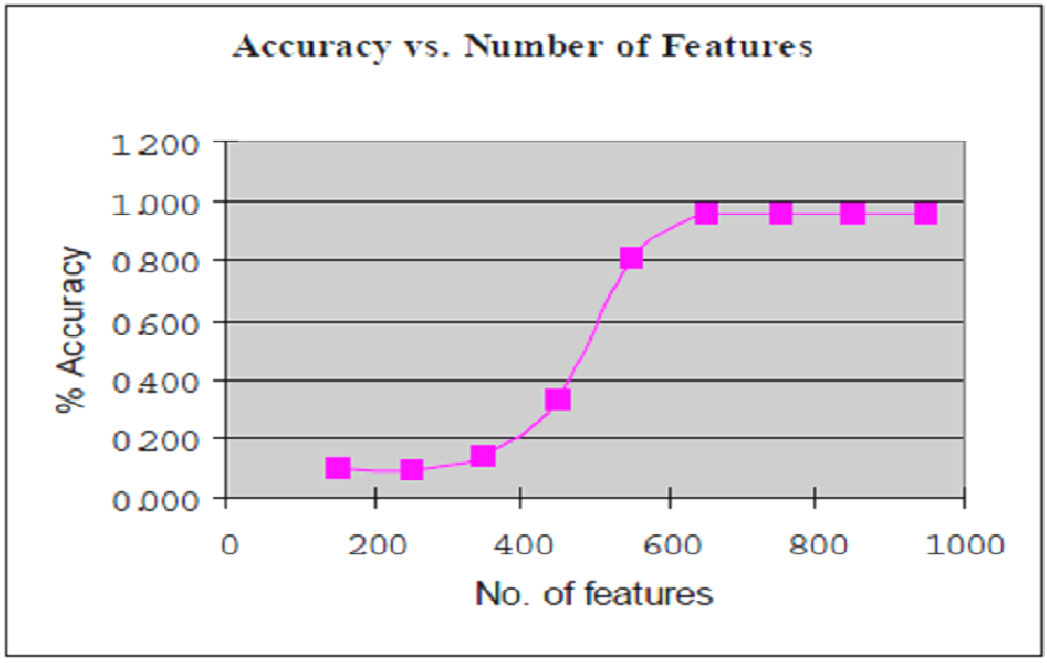

Fig. 5: The Classification accuracy with the PCA feature vectors

4.5 Class Prediction Phase and Results

The final phase of the classification of web pages is to classifying the new documents based on the error back propagation. After the preprocessing and applying the PCA on the stored documents in the database we get the dimensionality of feature vectors to be as an input to the neural networks. Therefore the PCA has been used to reduce the original feature vectors; here we have selected some different values for d means for PCA like (100, 200…, 600) and we choosing this parameter performs better for the web page classification compares with the other parameters to be input to the neural networks. The loading factor graph accumulated proportion of the Eigen values of principal components generated by the PCA is shown in the Fig. 4.

The error back-propagation neural networks parameters that have been used in the experiments based on the neural network PCA are shown in Table 1.

Table 1: The error back propagation Parameter

NN Parameters	Values
Learning rate (η)	0.05
Momentum rate (α)	0.001
Number of iteration (t)	1000
Means square error (MSE)	0.005

650 features have selected from the PCA and. That has found that this combination is the best in getting the accuracy of the classification as shown in Fig. 5.

650 input layer is use and 50 in the hidden layers and it is clear the output will compares with the predefined classes which we have 12(12 nodes in output layers) as a predefine classes as show in the table 2. The 50 hidden layers come from the try and error approach has been used to select the appropriate value of hidden layers which indicate the highest value of document classification accuracy, Fig. 6 shows the output of classification by the curves.

The principle of error back propagation is actually quite easy to understand, even though the math's behind it can Look rather daunting. The basic steps are:

1. Initialize the network with small random weights.

2. Present an input pattern to the input layer of the network.

3. Feed the input pattern forward through the network to calculate its activation value.

4. Take the difference between desired output and the activation value to calculate the network's activation error.

5. Adjust the weights feeding the output neuron to reduce its activation error for this input pattern.

6. Propagate an error value back to each hidden neuron that is proportional to their contribution of the network's activation error.

7. Adjust the weights feeding each hidden neuron to reduce their contribution of error for this input pattern.

8. Repeat steps 2 to 7 for each input pattern in the input collection.

9. Repeat step 8 until the network is suitably trained.

Table 2: The number of documents that have been used for classification.

Class no	Class name	No. of Docs
1	Base Ball	9
2	Boxing	2
3	Cycling	1
4	Football	8
5	Golf	4
6	Hockey	2
7	Motor Sports	1
8	Rugby	1
9	Basket Ball	2
10	Soccer	1
11	Cricket	10
12	Tennis	2
Total no. of Docs		43

It is important to note that each pattern is presented in turn, and the network adjusted slightly, before moving on to the next pattern. If we simply let the network perfectly correct the errors before moving onto the next pattern, it would never learn a generalized solution for the entire input collection.

The magic really happens in step 6, which determines how much error to feed back to each hidden neuron. Once the error value has been established.

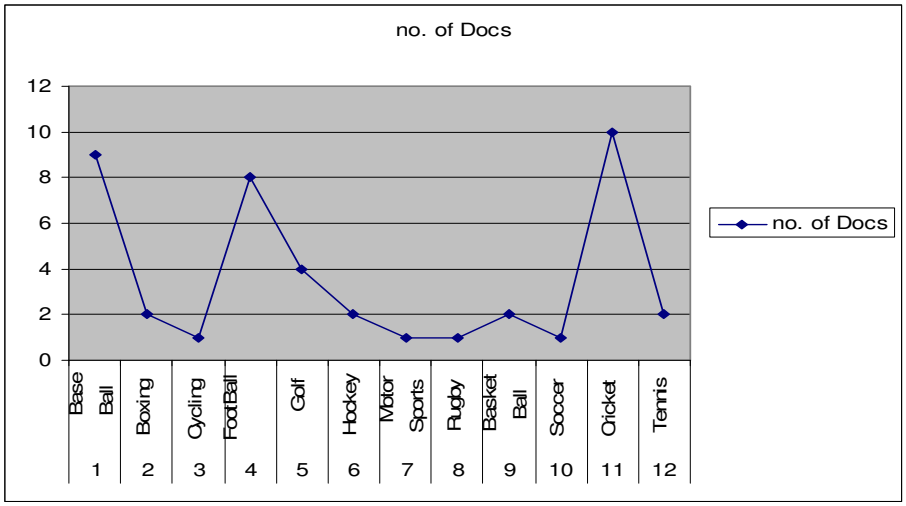

Fig. 6: the output result by curves of the classification

5.Conclusions

1- Web page classification is a type of supervised learning problem that aims to categorize web pages into a set of predefined categories based on labeled training data. Classification tasks include assigning documents on the basis of subject, function, sentiment, genre, and more. Unlike more general text classification, web page classification methods can take advantage of the semi-structured content and connections to other pages within the Web. Due to the large number irrelevant documents retrieved from different search engines. The classification provides facilities for web users specially organization institutes… etc, to select the specific documents.

2- The proposed system offers fast method for classification due to the summarizing of time by reducing the dimension and features of the word, also removing stop words will save spaces for storing document contents and reduce time taken during the search process.

3- This research showed that neural network approach improved accuracy in text classification task and is substantially better than other classifiers or neural network initialized randomly text classifiers performance comparable to other classifier results.

References

Blog, (2013). Available from: http://blog.peltarion.com/2006/07/10/classifier-showdown [Accessed 10th February 2013]

BMVA (2013). Available from: http://www.bmva.ac.uk/bmvc/1997/papers/duin/node5.html [Accessed 9th February 2013]

Chakrabarti, S. (2003) *Mining the Web Discovering Knowledge From Hypertext Data*. USA: Morgan Kaufmann Publishers, Elsevier Science

Fahmi, I, (2004a) *Examining Learning Algorithms For Text Classification In Digital Libraries.* Submitted In Partial Fulfillment of The Requirements for The Degree of Master of Arts at University of Groningen Groningen, The Netherland.

Fahmi, I, (2004b) *Examining Learning Algorithm for Text Classification in Digital Libraries*. Submitted in Partial Fulfillment of the Requirments for the degree of Master of Arts at University of Groningn, The Netherland.

Ipeirotis, P. G. (2004) *Classifying and Searching Hidden-Web Text Databases*. Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Graduate School of Arts and Sciences, Columbia University.

Jo, T. (2008) Neural Text Categorizer for Exclusive Text Categorization. *Journal of Information Processing Systems*, 4(2), p. 7.

Miltsakaki, E. and Troutt, A. (2009) Real-Time Web Text Classification and Analysis of Reading Difficulty. In:ISBI'09 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 28 June - 1 July, 2009, USA, Pages 89-97.

Minaei-Bidgoli, B. and Punch III W.F. (2003) Using Genetic Algorithms for Data Mining Optimization in an Educational Web-based System. *Part of the Lecture Notes in Computer Science book Sereis (LNCS, Volume 2724).*doi: 10.1007/3-540-45110-2_119 [Accessed 1st February 2013].

Mora-Jime´nez, I. and Figueiras-Vidal, A.R. (2009) Improving performance of neural classifiers via selective reduction of target levels.*Neurocomputing Journal,* 72(13-14), pp.2729-3410.

Resample (2013a). Available from: http://www.resample.com/xlminer/help/NNC/NNClass_intro.htm [Accessed 12th February 2013].

Resample (2013b). Available from: http://www.resample.com/xlminer/help/NNC/NNClass_intro.htm [Accessed 2nd February 2013].

Scime, **A.** (2005) *Web Mining: Applications and Techniques*. USA: Idea Group Publishing

Weingessel, A. and Hornik, K. (2000) Local PCA Algorithms, *IEEE Transactions on Neural Networks*, 11(6), p. 4.

Wikipedia (2013a). *Principal component analysis.* Available from: http://en.wikipedia.org/wiki/Principal_component_analysis [Accessed 1st February 2013].

Zhang, G. P. (2000) Neural Networks for Classifications: A Survey*, IEEE Transactions on Systems, Man, And Cybernetics—Part C: Applications And Reviews*, 30(4), p. 7.