Comparison of Novel Semi supervised Text classification using BPNN by Active search with KNN Algorithm

Mahak Motwani
[1]Assistant Professor, Computer Science Department, TCST
Bhopal,M.P.,462062 , India
mahak.motwani@trubainstitute.ac.in

Aruna Tiwari
[2]Assistant Professor, Computer Science Department, IIT
Indore, M.P, India
artiwari@iiti.ac.in

Abstract

With the availability of huge amount of text in internet, news, institutes, organization etc need of automatic text classification also increases, The proposed work comprised to deal with the major challenge of getting labeled data for training in classifier, since the availability of labeled data is expensive, time consuming, it also requires the involvement of annotator . A novel semi supervised test classification algorithm based on Back Propagation Neural Network is proposed which makes use of web assisted unlabeled data by Active search, this algorithm is compared with standard KNN algorithm on test data and standard data Mini Newsgroup. Experimental results state that the proposed algorithm outperforms KNN with Micro averaged F_1measure.

INTRODUCTION

A Major issue in the field of text classification is to organize large amount of documents into a number of meaningful classes. Text classification has application in the field of security, Bio medical, Company Resource Planning [1]. In existing Algorithm of Text classification Documents are represented using Vector space model which treats document as bag of words. Text representation is one of the crucial steps of text classification.

Depending on the data available text classification can be categorized as supervised or unsupervised. Supervised learning is learning from labeled examples. It is an area of machine learning that has reached development, it has generated general purpose and practically successful algorithms[2], whereas learning without the use of samples or labeled data is unsupervised learning where one finds an interesting structure with sample independently drawn from unknown distribution, Unsupervised learning is closely related to the problem of density estimation in statistics[3], major problems of unsupervised learning are minimum domain knowledge, noisy data, insensitive to instance order etc.

The large quantities of data is required to obtain high accuracy, and the difficulty of obtaining labeled data led the Research direct towards field where one can use a lot of unlabeled data which are easily available rather than labeled data which are manually assigned by experienced analyst which makes it time consuming and labor intensive job.

Semi supervised is a way to make use of this huge amount of easily available unlabeled data and few labeled data which makes it perform better than unsupervised algorithm, our proposed algorithm is making use of only few root words and easily available relevant data to train the classifier.

The Novel approach to text classification starts with just few root words using active search we collect web assisted data that undergoes the pre-processing , efficient text representation technique is used followed by BPNN, Our algorithm is compared with standard KNN on the basis of Micro averaged F_1measure.the rest of the paper is structured as follows. In Section 2 the Pre processing steps are described along with the term weighting method. Section 3 proposes the algorithm and comparison with KNN. Section 4 depicts the experimental Methodology and results. Section 5 concludes the research and discusses future prospects.

2. Pre processing in Text classification

2.1 Tokenization

Text document is a collection of sentences. In order to extract all words that are used in a given text, a tokenization [4] process is required for converting text document into stream of words by removing all punctuation marks such as commas, spaces, tabs, special characters etc, all text documents are merged to obtain set of different words which are collectively called the dictionary of a document collection.

2.2 Filtering Stop words

Filtering [5] is a method to remove words from the dictionary and thus from the documents. A standard filtering method is stop word filtering. The set of different words i.e. dictionary which is output of tokenization phase is now taken as input for the stop word filtering. The idea of stop word filtering is to remove words that have little or no content related information, like articles a ,an, the, conjunctions and, but, prepositions on, above, etc. It reduces complexity without any loss of information for typical application

2.3 Stemming

A stem is a natural group of words with equal meaning. (Andreas Hotho; 2005) [5]. Stemming method identifies the root of words for example run is the root word of running and ran. This methods try to construct the basic forms of words i.e. to remove 'ing' from verbs, plural's' from nouns, 'ed' from past tense or other affixes. A well-known rule based stemming algorithm has been originally proposed by Porter [Por80] [6]. He defined a set of production rules to iteratively transform (English) words into their stems. Each document words are preprocessed using Porter's stemming algorithm. After the stemming process, every word is represented by its stem.

2.4 Supervised Term weighting Method based on Relevance Factor Term Weighting Methods

The term weighting methods assigns an appropriate weight to the term to improve the performance of text classification[7] paper investigates several widely used unsupervised and supervised term weighting methods, a new simple supervised term weighting method, tf,rf, (term frequency, relevance frequency)is used to improve the terms' discriminating power for text categorization task, here emphasis has been made on term discriminating power analysis ,relevance factor refers to the degree of relevance of the term to the category it belongs to as compared with its relevance to other documents. It has been proved that it has a consistently better performance than other term most widely used term weighting methods Term frequency [8].

In text classification of multiple classes, a term may have high term frequency (t.f) and may belong to almost all the classes in this case the term actually do not posses a high discriminating power and so the inverse term document frequency factor and its variant has been used, our proposed algorithm uses a supervised term weighting method which is a multiplication of t.f and relevance factor r.f. where relevance factor is defined as

$$r.f = \log (2 + (a/\max (1, c)) \quad (1)$$

Here

a: total number of document in the positive category that contain this term

c: number of document in the negative category that contain this term

Here we assign a term as positive category if it belongs to the document that belongs to the category and all other categories combined together as negative category

3. Proposed Work

3.1 Proposed Algorithm

Labeling large amount of text spans for training systems is time consuming and unrealistic for many applications. We consider here the use of semi-supervised techniques, which lets to train a system with only a few labeled documents together with large amounts of unlabeled documents, It is difficult to build reliable classifier that is able to achieve high classification accuracy with of small number of available labeled documents, one way to overcome this problem is by using active search.

Active search is a way to first identify a number of important keywords, root words belonging to different category and then utilize search engines to retrieve from the web a multitude of relevant documents [9], we use Google to get relevant documents.

Though initially we have unrelated keywords, query word the web data or document collected will undergo effective Preprocessing and feature selection term weighting method to remove the irrelevant words and proceed for training. This data undergoes through the pre processing method of tokenization, stop word removal , application of porter stemming , we reduce the dimension by considering only those words that appear in more than one document usually words appearing in only one document has its correlation with that document example names ,such words do not specifically have discriminating power, such word are not considered.

Our Algorithm applies Supervised Term weighting Method based on Relevance Factor, it posses high discriminating capability of text words to the category. This data is fed to Neural Network classifier based on Back Propagation Neural Network. One of an efficient and popular approach for text categorization is Neural network, it can handle linear and nonlinear problems for text categorization, and both of linear [10] and nonlinear [11] classifier can achieve good results. There have been different neural networks applications to text categorization. Perceptron is the earliest and simple form of neural networks, which has only one input and an output layer, Ng, Goh, and Low first used the perceptrons to construct a text classifier, and reported a

surprisingly high performance [12]. Nakayama and Shimizu developed a training procedure for subject categorization using multilayer perceptrons [13]. The nonlinear neural networks are the more sophisticated neural networks with some hidden layers between the input and the output layers. Ruiz and Srinivasan compared the back propagation learning mechanism and counter propagation learning mechanism[14]. Back propagation neural network (BPNN) is the most popular in all of the neural network applications. It has the advantages of yielding high classification accuracy.

Back Propagation Neural Network based Classifier

Multilayer feed forward network which uses a supervised learning method, a generalization of delta rule is known as back propagation learning algorithm .Back propagation neural network. The training of a network by back propagation involves three stages: the feed-forward of the input training pattern, the calculation and back-propagation of the associated error, and the adjustment of the weight and the biases.

Input pattern feed-forward. Calculate the neuron's input and output. For the neuron j, the input Ij and output Oj are

$I_j=\sum W_{ij} * O_{j;}$ (2)

$O_j=f(I_j+ \theta_j)$ (3)

where wij is the weight of the connection from the ith neuron in the previous layer to the neuron j, $f(I_j+ \theta_j)$is an activation function of the neurons, Oj is the output of neuron j, and θ_j is the bias input to the neuron. In this paper, we use a tanh(n)sigmoid activation function defined with the equation:

tansig(n) = 2/(1+exp(-2*n))-1; (4)

This function is a good trade off for neural networks. The error, E, is calculated in this paper, the mean absolute error function is used in the output layer The mean absolute error is used to evaluate the learning effects and the training will continue until the mean absolute error falls below some threshold or tolerance level.

$E = \frac{1}{2\pi}\sum_n \sum_l \sqrt{(T_{nl} - O_{nl})^2}\, q$ (5)

Here n is the number of training patterns,l is the number of output nodes, and O_{nl} and T_{nl} are the output value and target value ,respectively. The mean absolute error is used to evaluate the learning effects and the training will continue until the mean absolute error falls below some threshold or tolerance level. The back propagation errors both in the output layer, δ_l and the hidden layer, δ_j , are then calculated with the following formulas:

$$\delta_l = \lambda (T_l - O_l) f'(O_l)$$

$$\delta_j = \lambda \sum_i \delta_l W_{ij} f'(O_j)$$

(5) (6)

Here T_l is the desired output of the l_{th} output neuron, O_l is the actual output in the output layer, O_j is the actual output value in the hidden layer, and k is the adjustable variable in the activation function. The back propagation error is used to update the weights and biases in both the output and hidden layers.

Weights and biases adjustment : The weights, wji, and biases, θi, are then adjusted using the following formulas:

$$W_{ji}(K+1) = W_{ji}(k) + \eta \delta_j O_i \quad (7)$$

$$\theta_i(k+1) = \theta_i(k) + \eta \delta_i \quad (8)$$

Here k is the number of the epoch and g is the learning rate.

The back propagation error is used to update the weights and biases in both the output and hidden layers.

3.2 KNN Algorithm

KNN algorithm

KNN is also known as instance based learning algorithm, Nearest Neighbor classifier are based on learning by analogy that is by comparing a test data with training data that is similar to it[15], after preprocessing each text document is now represented as a set of words and its corresponding numerical value specifying weightage of that term in document.

KNN algorithm [16] is a stable and efficient method of classification based on examples. Using KNN algorithm the process of document classification are as follows: In document set, we find the most similar K training documents for one given test documentation d. Then give each document class a value that is the similarity sum between the test documentation and the documentation in the K training documents belonging to the class. That is to say, if there are some documentation belonging to this class in the K documents, the value of this class is

the similarity sum between these documentation and the test documentation. Sorting by scores after getting the statistical value of the classes contain the K documents, we just consider the score more than threshold. Specific steps are as follows:

1) Assume K = the nearest number;

2) Calculate the similarity between the test documentation d and all training text;

3) Choose K documents, which is the most similar to the documentation d, as the nearest text of the documentation d.

4) Collect these classes of the nearest documents that have been choose.

5) Give each class a value based on the nearest K documents;

$$score(d, c_j) = \sum_{d_j \in KNN} sim(d, d_j) y(d_j, c_i) - b_i$$

$$y(d_j, c_i) = \left\{ \begin{matrix} 1\, d_{j \in} c_j \\ 0\, d_{j \notin} c_j \end{matrix} \right\}$$

bi is threshold.

6) The class with the biggest value is the class of the test document.

4.1Experimental Methodology

34 Query words of Computer science field,39 Query word of Medicine and 42 Query words of Sports are used to retrieve 100 documents of the three fields that are divided in 70 to 30 ratio of training and test documents, This training documents undergo Tokenization i.e removal of special character, numeric values etc, after tokenization we get 363985 terms from 210 training text files, followed by removal of 428 stop words, porter stemming algorithm is applied the terms reduces to 16768 words are retrieved that are filtered on the basis of occurrence in number of document, we retrieve only those terms that occur in more than one document and thus 6458 terms are considered.

Term weighting method based on relevance factor is used for feature representation, this data belonging to three category computer science ,medicine and sports undergoes training in BPNN with following parameters The parameters used for BPNN are neurons in input layer and 20 neurons in hidden layer, training function used is gradient descent adaptive training function, tansig as activation function for hidden layer and linear function for output layer, learning rate used is 0.3, momentum of 0.6.

4.2 Evaluation Criteria

Precision and Recall are two popular performance measures for text classification, precision is the fraction of retrieved documents that are relevant, recall is the fraction of relevant documents that are retrieved. The set of documents that are both relevant and retrieved is denoted as relevant ∩ retrieved,

Precision= Relevant ∩ retrieved/retrieved (9)

Precision =true positives/(true positives + false positives) (10)

Recall: this is the percentage of document that are relevant to the documents that are relevant to the query and were in fact, retrieved. it is formally defined as

Recall= relevant ∩ retrieved/relevant (11)

Recall = true positives/(true positives + false negatives) (12)

However, neither precision nor recall makes sense in isolation from each other as it is well known from the IR practice that higher levels of precision may be obtained at the price of low values of recall. To combine precision and recall, the two most widely used measures, i.e micro-averaged F_1 and macro-averaged F_1measure the F_1 measures are harmonic mean of Respective Precision and Recall.

Micro-averaged values are calculated by constructing a global contingency table and then calculating precision and recall using these sums. In contrast macro-averaged scores are calculated by first calculating precision and recall for each category and then taking the average of these. The notable difference between these calculations is that micro-averaging gives equal weight to every document (it is called a *document-pivoted measure*) while macro-averaging gives equal weight to every category (*category-pivoted measure*).

$$P_{\text{micro}} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + FP_i}; \quad R_{\text{micro}} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + FN_i} \qquad (13)$$

$$P_{macro} = \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{TP_i}{TP_i + FP_i}; \quad R_{macro} = \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{TP_i}{TP_i + FN_i} \tag{14}$$

4.3 Experimental Results

The proposed algorithm is implemented using MATLAB version 2012 MATLAB is a high-level language and interactive environment for numerical computation, visualization, and programming. The MATLAB neural network toolbox provides a complete set of functions and a graphical user interface for the design, visualization, implementation, and simulation of neural networks ,it is checked with the test data as well as on standard mini newsgroup data and compared with KNN algorithm.

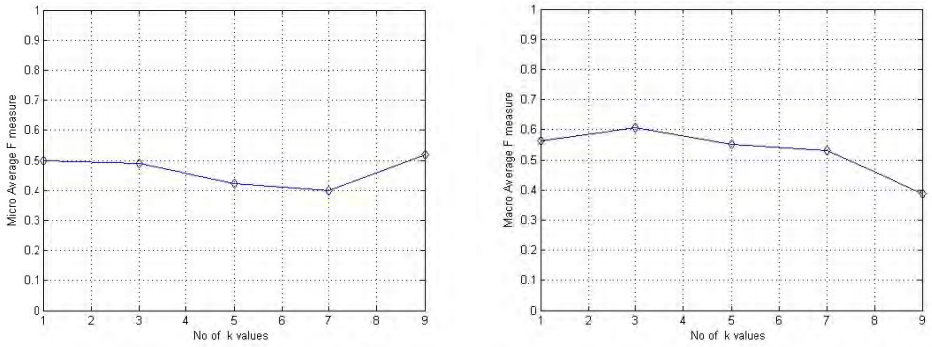

Figure 1,2. shows the performance of KNN on test data for different values of K on the basis of Micro F_1measure and Macro F_1measure, experimental results depicts that Knn performance is best for value k=3 which is 0.5 MicroAverageF_1measure and 0.60 MacroAverage F_1measure

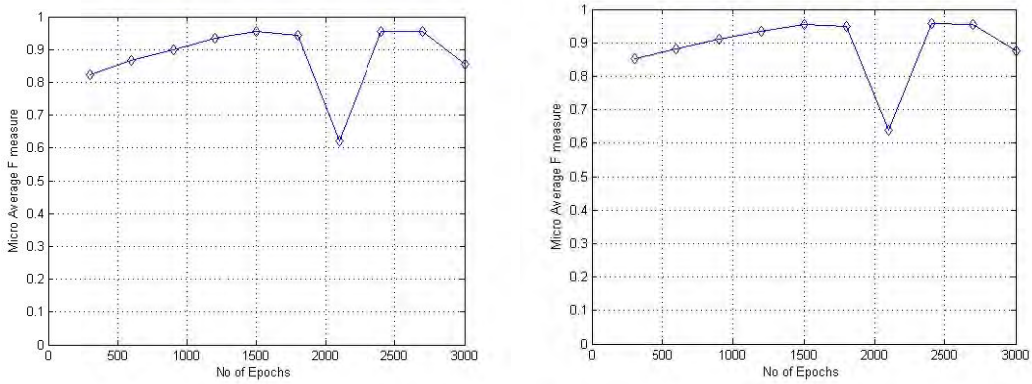

Figure 3,4 shows the performance of proposed algorithm on the basis of Micro Averaged and Macro Averaged F_1 measure on test data for different number of epochs, Proposed algorithm gives quite good results for test data, results are best MicroAverageF_1measure 0.955, MicroAverageF_1measure 0.958 at 2400 epochs

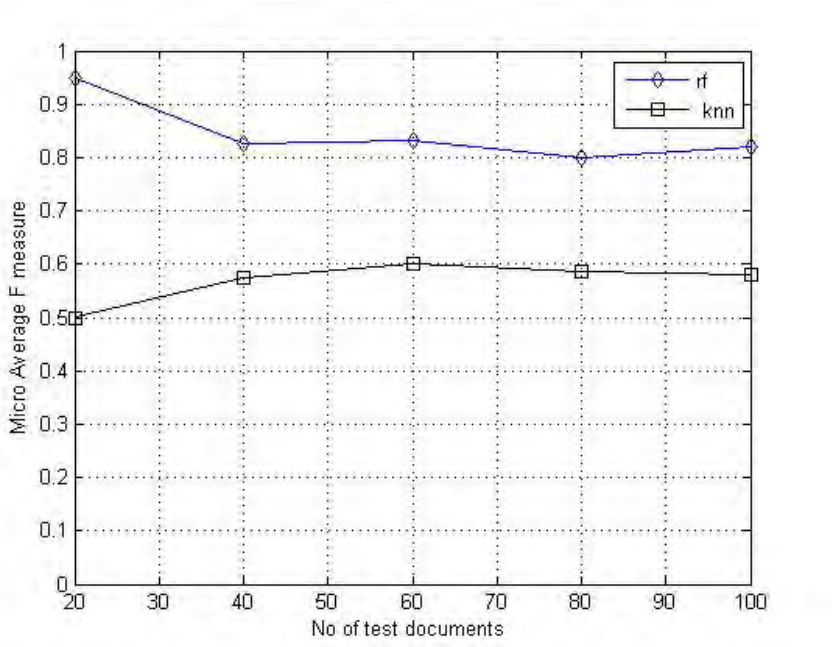

Figure 3 shows the performance of proposed algorithm and KNN on the basis of Micro Averaged F_1 measure on standard Mini Newsgroup data on random sets of 10 to 50 documents of computer and sports category of standard Mini newsgroup.

5. Conclusion and Prospects

The proposed work has been initiated to address various problems identified in the field of Text Classification i.e. unavailability of labeled documents, Using Active search , few numbers of keywords are used to get the relevant data of the categories of Computers, sports and Medicine. Efficient supervised Term weighting method based on Relevance factor is used for Text representation, this input is fed to BPNN, the algorithm output is calculated on different set of test data and standard mini newsgroup data with KNN algorithm on the basis of Micro Averaged measure F_1measure.It has been found that proposed algorithm outperforms KNN algorithm. Improvement in training time could be done by modifying BPNN.

References

[1] Falguni N. Patel, Neha R. Soni" Text mining: A Brief survey"International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970) Volume-2 Number-4 Issue-6 December-2012 243
[2] Maria-Florina Balcan, Avrim Blum,2010"A discriminative model of semi supervised learning"ACM DOI 10.1145/1706591.1706599
[3] Jordan, Michael I.; Bishop, Christopher M. (2004). "Neural Networks". In Allen B. Tucker. Computer Science Handbook, Second Edition (Section VII: Intelligent Systems). Boca Raton, FL: Chapman & Hall/CRC Press LLC.
[4] Andreas Hotho,Andreas N rnberger, Gerhard Paaß, "A Brief Survey of Text Mining", May 13, 2005
[5] Vallikannu Ramanathan, T. Meyyappan, "Survey of Text Mining", in International Conference on Technology and Business Management, March 18-20, 2013.
[6] M. Porter. An algorithm for suffix stripping. Program, pages 130–137, 1980
[7] Lan M,Tan C L,Su J,Lu Y, Supervised and traditional term weighting methods for automatic text categorization, IEEE Trans Pattern Anal Mach Intell. 2009 Apr;31(4):721-35.
[8] Mahak Motwani, Aruna Tiwari" Comparative Study and Analysis of Supervised and Unsupervised Term Weighting Methods on Text Classification" International Journal of Computer Applications (0975 – 8887) Volume 68– No.10, April 2013
[9] Zenglin Xu, Rong Jin , Kaizhu Huang† Michael R. Lyu, Irwin King, Semi-supervised Text Categorization by Active Search, CIKM'08, October 26–30, 2008, Napa Valley, California, USA, ACM 978-1-59593-991-3/08/10.
[10] Ma L, Shepherd J, Zhang Y (2003) Enhancing text classification using synopses extraction. In: Proceeding of the fourth international conference on web information systems engineering, pp 115–124
[11] Savio LY Lam, Dik Lun Lee (1999). Feature reduction for neural network based text categorization, 6th international conference on database systems for advanced applications (DASFAA '99)
[12] Ng HT, Goh WB, Low KL (1997). Feature selection, perceptron learning, and a usability case study for text categorization. In: Proceedings of the 20th annual international ACM-SIGIR conference on research and development in information retrieval, pp 67–73
[13] Nakayama M, Shimizu Y (2003) Subject categorization for web educational resources using MLP. In: Proceedings of 11th European symposium on artificial neural networks, pp 9–14
[14] Ruiz ME, Srinivasan P (1998). Automatic text categorization using neural network. In: Proceedings of the 8th ASIS SIG/CR workshop on classification research, pp 59–72
[15] Eui-Hong (Sam) Han,George Karypis, Vipin Kumar "Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification", Advance in Knowledge Discovery and Data Mining,Lecture notes in computer science ,volume 2035, 2001 p 53-65
[16] Dempster A , Laird N , Rubin D. Maximum likelihood eitimation from incomplete data via EM algorithm [J] . J . Royal Statistical Society Series B , 1997 , 3 9 .