International Journal of Engineering Research and General Science Volume 2, Issue 6, October-November, 2014
ISSN 2091-2730

Document-Document similarity matrix and Naive-Bayes classification to web information retrieval

Dr.Poonam Yadav[1]

[1]*D.A.V College of Engineering. & Technology, India*
poonam.y2002@gmail.com

Abstract— Due to the continuous growth of web database, automatic identification of category for the newly published web documents is very important now-a-days. Accordingly, variety of algorithms has been developed in the literature for automatic categorization of web document to easy retrieval of web documents. In this paper, Document-Document similarity matrix and Naive-Bayes classification is combined to do web information retrieval. At first, web documents are pre-processed to extract the features which are then utilized to find document-document similarity matrix where every element within matrix is similarity between two web documents using semantic entropy measure. Subsequently, D-D matrix is used to create a training table which contains the frequency of every attributes and its probability. In the testing phase, relevant category is found for the input web document using the trained classification model to obtain the relevant categorized documents from the database. The relevant category identified from the classifier model is used to retrieve the relevant categorized documents which are already stored in the web database semantically. The experimentation is performed using 100 web documents of two different categories and the evaluation is done using sensitivity, specificity and accuracy.

Keywords— Information retrieval, Naive-bayes classification, semantic retrieval, web document categorization, D-D matrix, accuracy, specificity.

1. INTRODUCTION

With the ever seen growth of web database, relevant information retrieval finds major difficulties in most of the time because of extensive availability information and lack of effective approaches. To retrieve the information effectively and efficiently, automatic categorizing of web document is important [1] for information retrieval system. We know that, information stored in the web database is growing continuously so when new information is published in web database, retrieving those information if it is most relevant category is also important for user [2-6]. In order to accomplish this task, automatic identification of category for a new web document is definitely needed in current days. With the aim of this, classification-based algorithms [11-13] are proposed in the literature for automatic categorization of web documents. For example, K-NN [7-10], naive bayes classifier and adboost algorithms are benchmark algorithm utilized by various researchers for classification.

In this paper, web information retrieval for an input query document is done using document to document similarity matrix and naive bayes classifier. At first, input web document are converted to feature space using a set of pre-processing techniques. Then, D-D matrix is constructed using semantic entropy measure which considers multiple considerations mathematically. The similarity space is given to naive bayes classifier [15] to construct training table. Finally, relevant category of the input web documents is found out using testing phase of naive classifier. The relevant category can easily output the relevant web document as an output to the user. The paper is organized as follows: Section 2 presents K-NN classifier and section 3 presents the proposed algorithm for information retrieval. Section 4 presents the experimental result and finally, conclusion is given in section 4.

2. K-NN CLASSIFICATION FOR WEB INFORMATION RETRIEVAL

K-NN [7-10] is one of the standard algorithms for classification which is the process of identifying a relevant group or class for any input data. K-NN classification can be done using three important steps. In the first step, distance is found out for a query data with all of the training data available in the training database. In the second step, most similar k-number of data is identified through the minimum distance. Lastly, class label of the query data will be identified from k-number of similar data through majority voting.

Drawbacks: When taking K-NN classification algorithm for web document categorization, two important challenges should be handled. The first challenge is how to identify the similarity among the documents with the inputting document. The second challenge is how to avoid the similarity matching with the entire training document because matching the similarity with the entire web database

is very tough. In order to solve these two challenges, semantic entropy measure (SE) is used instead of Euclidean distance and naive bayes classifier is used instead of K-NN classification.

3. DOCUMENT-DOCUMENT SIMILARITY MATRIX AND NAIVE-BAYES CLASSIFICATION TO WEB INFORMATION RETRIEVAL

This section presents the proposed web information retrieval algorithm using document-document similarity matrix and naïve-bayes classifier. The block diagram of the proposed method is given in figure 1. The method is explained using two different phases.

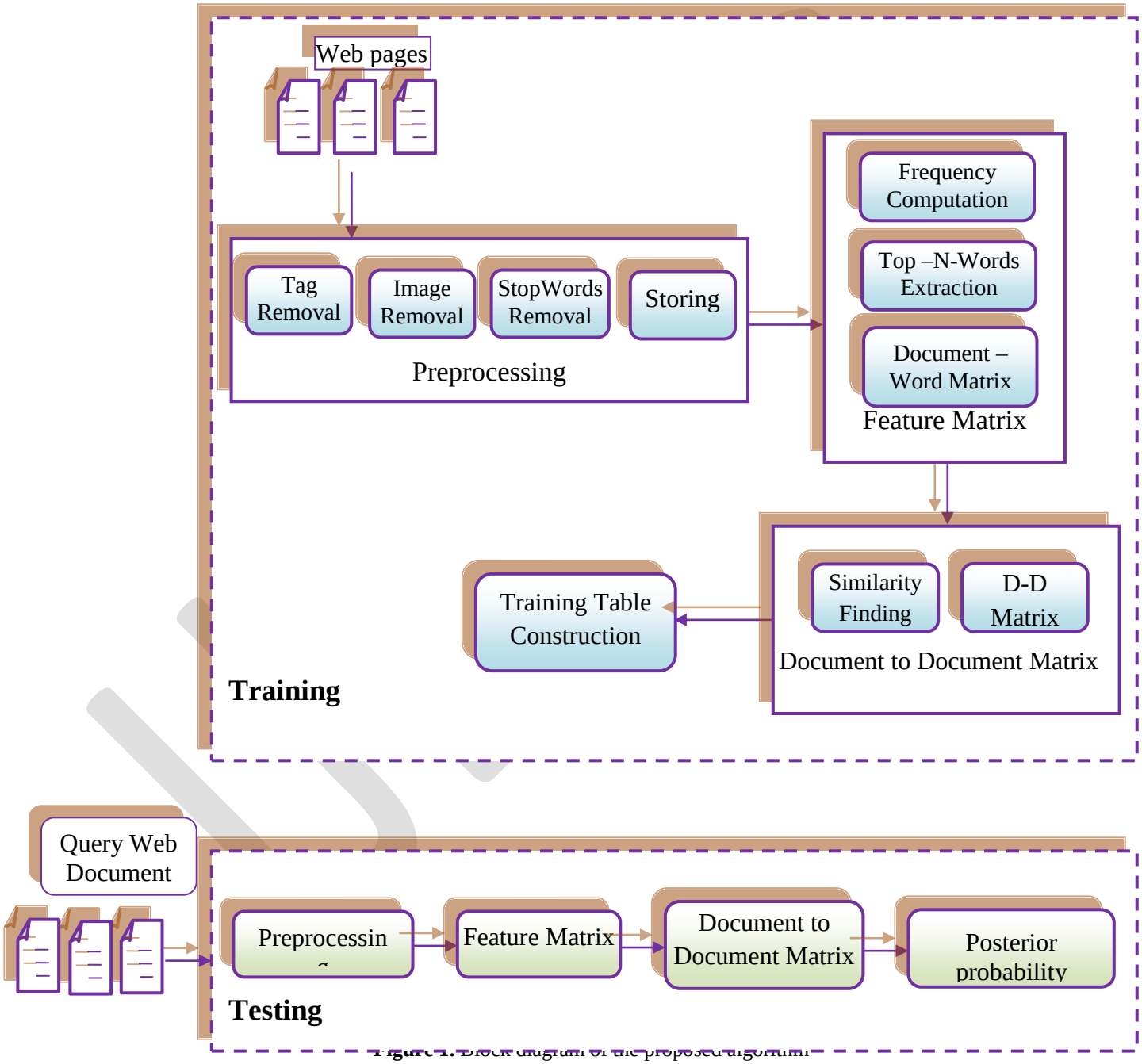

Figure 1. Block diagram of the proposed algorithm

In the first phase, web documents are pre-processed to extract the features which are then utilized to find document-document similarity matrix. The D-D matrix is used to create a training table which contains the frequency of every attributes and its probability. In the testing phase, a relevant category is found out using the classification model to obtain more relevant categorized documents from the web database.

3.1 Preprocessing

The input web database W which has m number of web documents is taken as input along with its relevant category. For every web document D_i, pre-processing is applied to extract the relevant keywords. In order to find the relevant words from web document, all the html tags are identified and removed from the web document. Once we remove tags and images, stop words such as, "can, could, is, was , may" are matched with pre-defined set to obtain only meaningful words. After obtaining meaningful words, root form of all the keywords is obtained to make all the derived format of words into its original format. For every words identified, frequencies are computed within web document and top-N words chosen as vector to represent the web page document.

3.2 Document-Document similarity matrix computation

The document vector obtained from the previous step is then given to D-D matrix computation process. This matrix is generated by finding similarity among all the web documents. The document to document similarity matrix is indicated as D-D matrix which is in the size of $m*m$. Every element within matrix is similarity between two web document having top-N extracted keywords. The similarity is computed based on measure, called semantic entropy measure (SE) [14]. Let us consider that d_1 and d_2 are two documents. The document d_1 have $k1$ number of keywords and document d_2 have $k2$ number of keywords. The unique keywords (m) are taken and frequencies of the keywords are represented in a vector d_{1_j}. Similarity, frequencies of unique keywords belonging to d_2 are represented in a vector d_{2_j}. f_{D_1} is frequency of the keywords in D_1, f_{D2} is the frequency of keywords in D_2, $\neg f_{D_1}$ represents the frequency of keywords in the synonyms set, $\neg f_{D_2}$ is the frequency of keywords in the synonyms set. Here, synonyms set are computed by giving the keywords of document to the wordnet ontology. Based on this assumption, the proposed SE-measure is formulated as,

$$SE_{measure} = -\left(P_r(D_1, D_2)\log P_r(D_1, D_2)\right) - \left(P_r(D_1, \neg D_2)\log P_r(D_1, \neg D_2)\right) - \left(P_r(\neg D_1, D_2)\log P_r(\neg D_1, D_2)\right) - \left(P_r(\neg D_1, \neg D_2)\log P_r(\neg D_1, \neg D_2)\right) \quad (1)$$

The values of $P_r(D_1, D_2)$, $P_r(D_1, \neg D_2)$, $P_r(\neg D_1, D_2)$ and $P_r(\neg D_1, \neg D_2)$ are defined as follows,

$$P_r(D_1, D_2) = \frac{1}{m}\sum_{i=1}^{m} 2\left(\frac{f_{D_1} + f_{D2}}{\max(f_{D_1}, f_{D_2})}\right) \quad (2)$$

$$P_r(D_1, \neg D_2) = \frac{1}{m}\sum_{i=1}^{m} \frac{f_{D_1}}{(f_{D_1} + f_{D_2})} \quad (3)$$

$$P_r(\neg D_1, D_2) = \frac{1}{m}\sum_{i=1}^{m} \frac{f_{D2}}{(f_{D_1} + f_{D_2})} \quad (4)$$

$$P_r(\neg D_1, \neg D_2) = \frac{1}{m}\sum_{i=1}^{m} 2\left(\frac{\neg f_{D_1} + \neg f_{D_2}}{\max(\neg f_{D_1}, \neg f_{D_2})}\right) \quad (5)$$

3.3 Naive-Bayes classification to web information retrieval

D-D matrix obtained from the training web document is given for the training process to construct the training table. Training table is utilized to find the category of test web document. The relevant category identified from the classifier model is used to retrieve the relevant categorized documents already stored in the web database. **Training:** Let assume that D-D matrix of training data contains

'm' number of attributes, d . Here, D-D matrix is segmented as different number of categories based on ground truth. After that, for every category of web documents, mean and variance is computed for every attributes. Assume that μ_c and σ_c^2 be the mean and variance of the D-D matrix belonging to every attributes of class c. Subsequently, the probability of attributes given in a class for the D-D matrix, $P(d = D - D|C)$, can be calculated using Gaussian distribution formulae with mean μ_c and variance, σ_c^2 . That is,

$$P(d = D - D \mid c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{\frac{-(d-\mu c)^2}{2\sigma_c^2}} \tag{6}$$

Based on the above formulae, training table of size, $C * m$ is constructed. Every element in this matrix is computed by finding the probability value belongs to category label with respect to attribute.

Testing: In the testing phase, input web document w_T is taken and its 'm' attribute value are found out by computing similarity between the training documents. Once we find 'm' attribute values, the category of test web document is calculated based on the objective,

$$classify\,(w_T) = \underset{c}{\operatorname{argmax}}\, p(C = c) \prod_{i=1}^{m} p(D_i = _d) \mid C = c). \tag{7}$$

$$posterior(c_i) = \frac{prob(c_i) \times \prod_{i=1}^{m} prob(c_i \mid d)}{evidence} \tag{8}$$

$$evidence = \sum_{j=1}^{n} prob(c_i) * \prod_{i=1}^{m} prob(c_i \mid d) \tag{9}$$

Semantic Retrieval: The posterior probability of every class for the input web document is computed and the posterior probability which is greater in the corresponding category is given as final category. Based on the category found, the categorized documents stored in the database are given as final output to user.

4. RESULTS AND DISCUSSION

This section presents experimental results and discussion of the proposed D-D matrix-based naïve bayes classifier.

4.1 Evaluation with sensitivity

The proposed D-D matrix-based naïve bayes classifier algorithm is implemented with 100 web documents having two groups, one is related with sports articles and other one is related with politics' related articles. Every group contains 50 documents and it is given as input to the algorithm. For training, 80% of the documents from each group is taken for building the training table and remaining 20% of document from every group is used as testing dataset. The obtained classification results are evaluated with sensitivity.

$$Sensitivity = TP/(TP + FN) \tag{10}$$

Where, TP stands for True Positive, TN stands for True Negative, FN stands for False Negative and FP stands for False Positive.

The performance plot of the proposed D-D matrix-based naïve bayes classifier algorithm and D-D matrix-based k-NN algorithm is given in figure 2. From the figure, we can easily understand that the proposed algorithm providing good sensitivity. The proposed algorithm reached of about 85% sensitivity as compared with existing algorithm reaches the value of 82%.

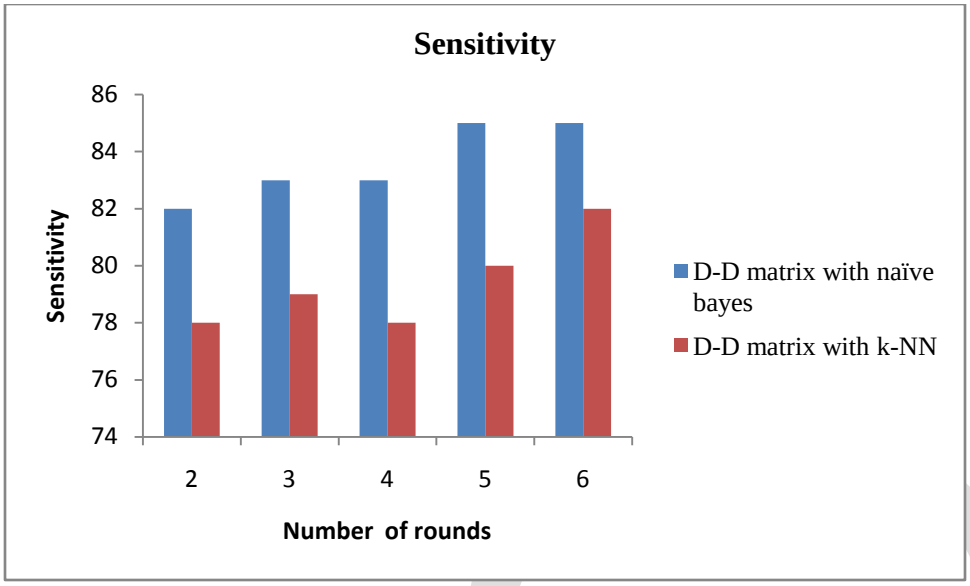

Figure 2. Sensitivity plot in between the proposed and existing

4.2 Evaluation with specificity

The proposed D-D matrix-based naïve bayes classifier algorithm is implemented with 100 web documents and the obtained classification results are evaluated with specificity

$$\text{Specificity} = \text{TN}/(\text{TN}+\text{FP}) \tag{11}$$

Where, TP stands for True Positive, TN stands for True Negative, FN stands for False Negative and FP stands for False Positive.

The performance plot of the proposed D-D matrix-based naïve bayes classifier algorithm and D-D matrix-based k-NN algorithm is given in figure 3. From the figure, we can easily understand that the proposed algorithm providing good specificity. The proposed algorithm reached of about 76% specificity as compared with existing algorithm reaches the value of 70%.

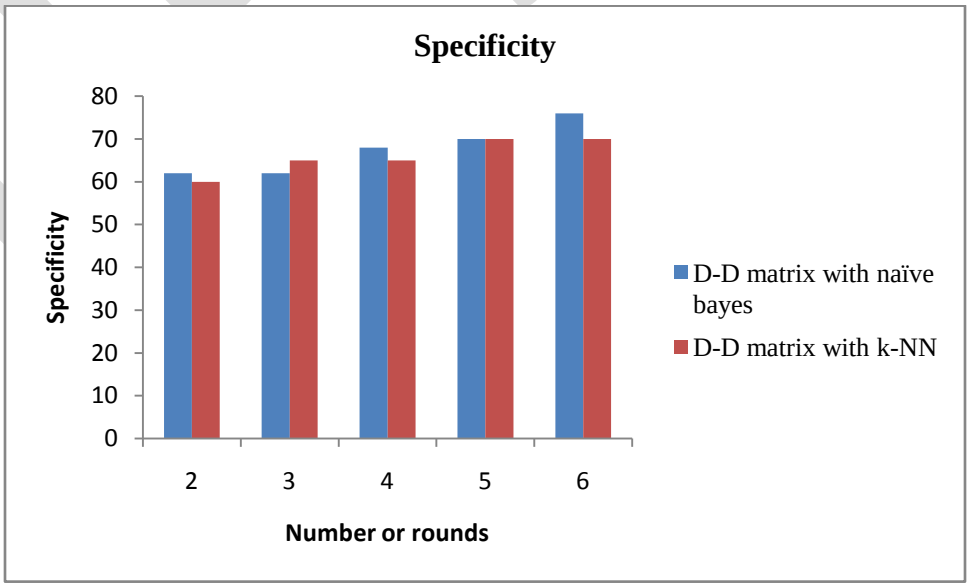

Figure 3. Specificity plot in between the proposed and existing

4.3 Evaluation with accuracy

The proposed D-D matrix-based naïve bayes classifier algorithm is implemented with 100 web documents having two groups, one is related with sports articles and other one is related with politics' related articles. The obtained classification results are evaluated with accuracy.

$$Accuracy = (TN + TP)/(TN + TP + FN + FP) \quad (12)$$

Where, TP stands for True Positive, TN stands for True Negative, FN stands for False Negative and FP stands for False Positive.

The performance plot of the proposed D-D matrix-based naïve bayes classifier algorithm and D-D matrix-based k-NN algorithm is given in figure 4. From the figure, we can easily understand that the proposed algorithm providing good accuracy. The proposed algorithm reached of about 76% accuracy as compared with existing algorithm reaches the value of 73%.

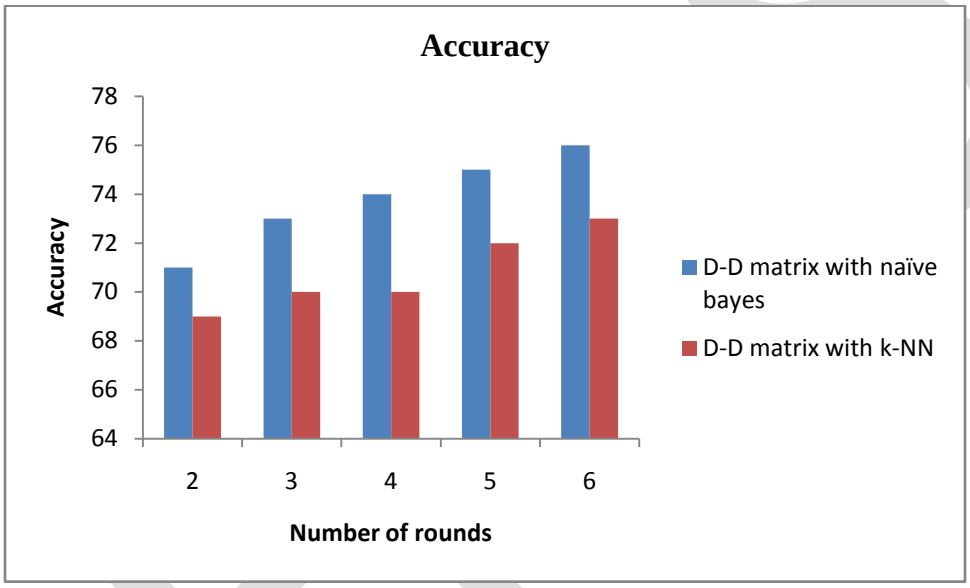

Figure 4. Accuracy plot in between the proposed and existing

5. CONCLUSION

In this paper, Document-Document similarity matrix and Naive-Bayes classification was combined to do web information retrieval. The semantic entropy measure was used to construct D-D matrix after performing pre-processing and feature construction. Then, Naive-Bayes classifier was utilized to find the relevant category and subsequently, the required information for the user. The proposed D-D matrix-based naïve bayes classifier algorithm was implemented with 100 web documents having two groups, sports articles and politics' related articles. The performance of the proposed algorithm was analyzed with sensitivity, specificity and accuracy. From the experimentation evaluation, the finding is that the proposed algorithm reached of about 85% sensitivity as compared with existing algorithm which reaches only the value of 82%. Also, the proposed algorithm reached of about 76% specificity and 76% accuracy as compared with existing algorithm.

REFERENCES:

[1] Ming Chen ; Hofestadt, R., "Web-based information retrieval system for the prediction of metabolic pathways", IEEE Transactions on NanoBioscience, Vol. 3, NO. 3, PP. 192 - 199, 2004.
[2] Killoran, J.B., "How to Use Search Engine Optimization Techniques to Increase Website Visibility", IEEE Transactions on Professional Communication, vol. 56, no. 1, pp. 50-66, 2013.

[3] Böhm, T, Klas, C.-P. ; Hemmje, M., "ezDL: Collaborative Information Seeking and Retrieval in a Heterogeneous Environment", computer, IEEE, vol. 47, no. 3, pp. 32-37, 2014.

[4] Sumiya, K., Kitayama, D. ; Chandrasiri, N.P., "Inferred Information Retrieval with User Operations on Digital Maps", IEEE Internet Computing, vol. 18, no. 4, pp. 70-73, 2014.

[5] Xiaogang Han, Wei Wei ; Chunyan Miao ; Jian-Ping Mei ; Hengjie Song, "Context-Aware Personal Information Retrieval From Multiple Social Networks", Computational Intelligence Magazine, IEEE, vol. 9, no. 2, 2014.

[6] Junnila, V., Laihonen, T., "Codes for Information Retrieval With Small Uncertainty", IEEE Transactions on Information Theory, vol. 60, no. 2, pp. 976-985, 2014.

[7] Jinn-Min Yang ; Pao-Ta Yu ; Bor-Chen Kuo, "A Nonparametric Feature Extraction and Its Application to Nearest Neighbor Classification for Hyperspectral Image Data", IEEE Transactions on Geoscience and Remote Sensing, vol. 48, no. 3, pp. 1279-1293, 2010.

[8] Zhao, S. Rui, C.; Zhang, Y., "MICkNN: multi-instance covering kNN algorithm", Tsinghua Science and Technology , IEEE, vol. 18, no. 4, 2013.

[9] Li Ma, Crawford, M.M. ; Jinwen Tian, "Local Manifold Learning-Based k -Nearest-Neighbor for Hyperspectral Image Classification", IEEE Transactions on Geoscience and Remote Sensing, vol. 48, no. 11, pp. 4099-4109, 2010.

[10] Aslam, Muhammad Waqar, Zhu, Zhechen ; Nandi, Asoke Kumar, "Automatic Modulation Classification Using Combination of Genetic Programming and KNN", IEEE Transactions on Wireless Communications, vol. 11, no. 8, pp. 2742-2750, 2012.

[11] Khabbaz, M. Kianmehr, K. ; Alhajj, R., "Employing Structural and Textual Feature Extraction for Semistructured Document Classification", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol. 42, no. 6, pp. 1566 - 1578, 2012.

[12] Xuan-Hieu Phan ; Sendai, Japan ; Cam-Tu Nguyen ; Dieu-Thu Le ; Le-Minh Nguyen, "A Hidden Topic-Based Framework toward Building Applications with Short Web Documents", IEEE Transactions on Knowledge and Data Engineering, Vol. 23, no. 7, pp. 961 - 976, 2011.

[13] Yajing Zhao ; Jing Dong ; Tu Peng, "Ontology Classification for Semantic-Web-Based Software Engineering", IEEE Transactions onServices Computing, Vol. 2, no. 4, pp. 303-317, 2009.

[14] Poonam yadav, "SE-K-NN classification algorithm for semantic information retrieval".

[15] Sang-Bum Kim ; Kyoung-Soo Han ; Hae-Chang Rim ; Sung Hyon Myaeng, "Some Effective Techniques for Naive Bayes Text Classification", IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 11, pp. 1457 - 1466, 2006