Research Science Press

International Journal of Data Analysis and Information Systems
VOLUME 8 • NUMBER 1 • JUNE 2016

journal homepage : serialsjournals.com

ISSN : 2229-5887

An Efficient Dynamic Clustering Method for Web Document Clustering

Soumen Swarnakar[1], Sreya Dutta[2], Sayani Sarkar[2], Kausani Chakroborty[2] and Saheli Kar[2]

[1]*Department of Information Technology, Asst. Professor, Netaji Subhash Engineering College, Techno City, Garia Kolkata-700152, India*

[2]*B.Tech Student, Department of Information Technology, Netaji Subhash Engineering College, Techno City, Garia Kolkata-700152, India*

Authors emails

soumen_swarnakar@yahoo.co.in

sreyadutta@yahoo.co.in

sayani.sarkar95@gmail.com

kausani.23@gmail.com

sahelikar067@gmail.com

ABSTRACT

The size of the web is gradually increasing with each passing day, thus internet has become the largest data repository nowadays. Thus it may face the problem of information overloading. In World Wide Web there exists abundant information with the mixture of dynamic and heterogeneous data. So it is very difficult for the average user to search and get the relevant information. It is necessary to develop a technique that can help the users to find the required information through the web in an optimized way. Web document clustering will help the users to search for a particular document in an enhanced way and the aim of this paper is to improve the efficiency and accuracy of dynamic web document clustering. The core idea of dynamic clustering is that the new object should always be clustered after comparing it with the existing clusters. In this paper an efficient dynamic web document clustering approach has been developed using ontology tree based concept similarity and hash indexing which eventually reduces the time for clustering of documents and improves the performance from the known standard document clustering methods. Standard cluster quality measures like F-Measure and purity have been used for measuring the quality of the clusters.

Keywords: *Dynamic, ontology, web document, clustering, hash indexing*

INTRODUCTION

The Internet has become the largest data repository of hypertext, although the web search environment is not perfectly ideal. Search engines try to match the query terms and the keywords that already exist in the World Wide Web. To achieve this objective one of the most popular techniques used is web document clustering. Document clustering involves the use of descriptors and descriptor extraction. Descriptors are sets of words that describe the contents within the clusters and are considered to be a centralized process. Nowadays, most people depend on the World Wide Web technology and it becomes a great challenge to organize the web documents in groups along with simultaneous arrival of new web documents and sorting them into existing groups. Using dynamic clustering procedure the web documents are either grouped within one of the existing clusters or it can form a completely new cluster, so that the documents have higher inter-dependency within a cluster and a lower intra-dependency among the clusters. Examples of document clustering include web document clustering for search users along with wide applications in various areas of web mining, search engines, and information retrieval methods.

LITERATURE REVIEW

Many clustering technique already exist in the literature (Han and Kamber, 2001). Yang Yan *et al.* (2012) introduced a new heuristic semi supervised fuzzy co-clustering algorithm (SS-HFCR) for categorization of large web documents. Xufei Wang *et al.* (2011) discussed on document clustering via matrix representation where rows represent distinct terms and columns represent cohesive segments that improves the cluster quality. Nora Oikonomakou *et al.* (2005) developed an exhaustive survey of web document clustering

approaches available on the literature, showing classification into three main categories: text based, link based and hybrid. Adam Schenker *et al.* (2004) presented a unique algorithm where the objects to be clustered are represented by graphs rather than the usual case of numeric feature vectors. R. Nagaraj et al. (2014) discussed about the correlation similarity measure based document clustering with directed ridge regression. In this paper the correlation preserving indexing and directed ridge regression are compared and then the experimental results are obtained when the number of nearest neighbours is set to seven or eight. B. NageswaraRao *et al.* (2014) provided a survey on document clustering and they introduced a new document clustering method based on correlation preserving indexing which maximizes the correlation between the documents inside the clusters and minimizes the correlation between the documents outside the clusters, although, a low dimensional semantic subspace is derived where the documents corresponding to the same semantics are close to each other. It reduces the computational cost. In the paper of Xiao Shen *et al.* (2003) there is also a wide range of discussion on the topic of algorithm of documents clustering based on minimum spanning tree. In this paper, portions of documents are given that are used in the experiments of Chinese Times news and when compared between the results of the average precision and precision-recall of the three methods - MST, k-means and single pass, it was proved that the "clustering quality" of MST method is better than the other two. Paper presented by Jing Peng *et al.* (2007) discussed on the clustering algorithm for short documents based on concept similarity, where a new text clustering algorithm–CACS has been proposed, which suggests a word similarity and document similarity formula based on concept similar relations. Shehata, S. (2010) discussed on concept-based mining model on different data sets in text clustering. Jayabharathy *et al.* (2012) proposed three dynamic document clustering algorithms, namely: Term frequency based MAximum Resemblance Document Clustering (TMARDC), Correlated Concept based MAximum Resemblance Document Clustering (CCMARDC) and Correlated Concept based Fast Incremental Clustering Algorithm (CCFICA) are proposed.

In web document clustering the main objective is to cluster the web documents. In this paper at first the web pages are prepossessed and then the words are organized using hash index to create ontology tree for different web documents in consultation with concept and the synonym dictionary easily. A new concept of creating indices of web document clusters has been put forward in comparison to the newly arrived document concept.

The paper has been divided into four sections. Section 2 describes the methodology containing proposed model, approach for dynamic clustering while Section 3 illustrates the analysis of the result and comparisons of performance with proposed method. The conclusions are summarized in Section 4.

METHODOLOGY

Some terminologies are defined below for ready reference to the paper.

Concept- For a particular domain different terms can create a concept. For example – LAN, network, internet, intranet etc. can create the concept of networking.

Dictionary- For web document clustering a concept dictionary is maintained where different related concepts and their related terms are stored. In this paper a synonym dictionary is maintained which stores synonyms of terms or objects. This synonym dictionary is used to create document vector. The example of concept dictionary and synonym dictionary are shown below.

i) Concept dictionary

Concept	Object
Networking	bridge, LAN, socket, intranet, network, internet
Database	schema, data, SQL, normalization, query
Operating system	scheduling, deadlock, process, thread

ii) Synonym dictionary

Object	Synonymous Terms
LAN	MAN, WAN

3.1 Proposed model

Figure 1 shows the sequences or steps involved in dynamic web document clustering.

3.2 Proposed Approach

In this paper a new and efficient dynamic web document clustering approach has been described. The steps involving in this approach have been described below:

Step 1: Pre-processing of web documents

Here the sentences are tokenized. Then all stop words, articles, prepositions, conjunctions are removed. After that, word stemming is done. i.e. say a word "injured" is present in the document, after stemming the word will become "injure".

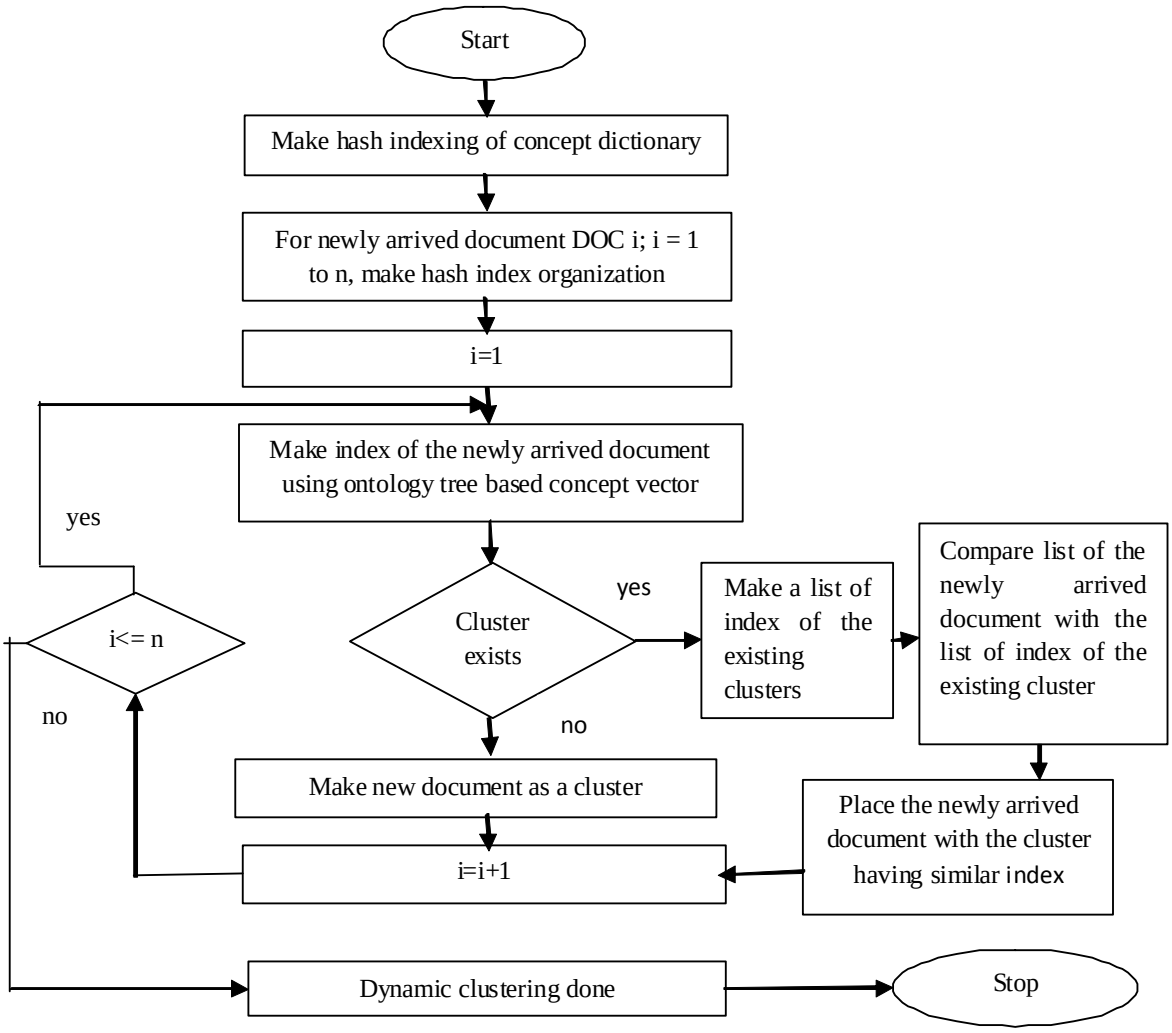

Figure 1: Steps involved in dynamic web document clustering

Step 2: Organize concept dictionary with hash index:

Because of fast searching purpose, here at first the concept dictionary has been organized using hash index. At first all objects under a concept have been sorted alphabetically and then the hash indexing is done along with the main concept. Suppose, the concept dictionary is as follows:

Concept	object
Networking	bridge, LAN, socket, intranet, network, internet
Database	schema, data, SQL, normalization, query
Operating- system	scheduling, deadlock, process, thread

For each concept the objects under that concept are sorted alphabetically and then they are hashed with starting alphabet. All objects belonging to a specific concept point are linked to the concept they belong to. Similarly hash index organizations for other concept dictionaries are also formed for each concept.

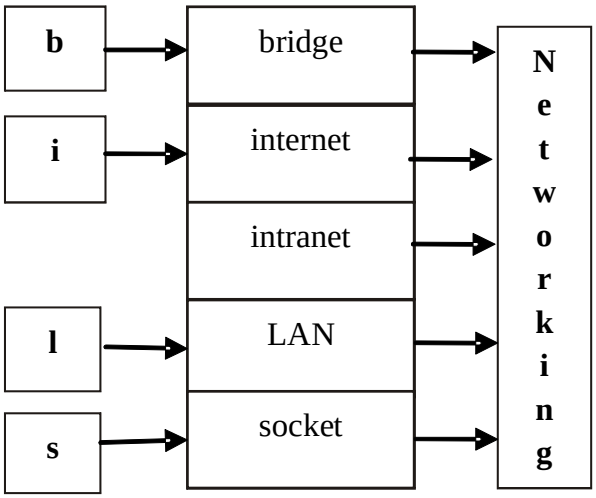

Figure 2: Hash index organizations of sorted objects with concept

In the above figure 2, all the objects under the networking concept are sorted alphabetically. Then they are hashed and they are pointing to one single concept that is networking.

In this same way, the synonymous terms of the synonym dictionary can also be sorted and indexed with hashing.

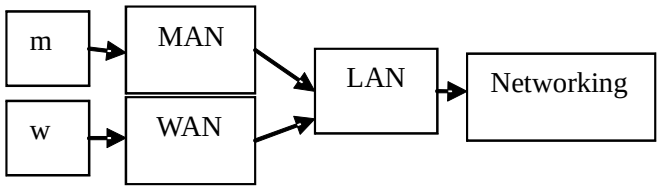

Figure 3: Hash organization of sorted terms of synonym dictionary

In the above figure 3, it has been shown that the terms MAN and WAN are synonymous to LAN. The network concepts are then sorted alphabetically and hashed with LAN. As LAN is an object of the Networking concept, a pointer from LAN is used to point the networking concept.

Step 3: Organize the terms of pre-processed web document using hash index

Words of each pre-processed web documents are sorted with hash index to search them in the minimum time possible. Suppose if the web document contains a line - "LAN is local area network" then after pre-processing, the terms {LAN, local, area, network} will be organized using hash organization as shown below in figure 4.

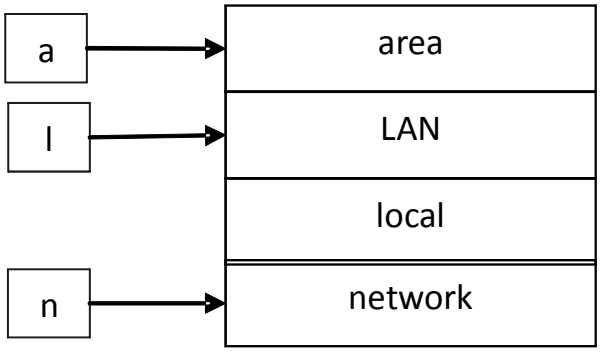

Figure 4: Hash organization of sorted terms of web document

Step 4: Ontology tree creation

Ontology is the formal explicit description of concepts while properties of each concept described by its attributes and stored as objects in the concept dictionary. Using hash index, an efficient searching of objects has been designed to create the ontology tree corresponding to web documents through consultation of the concept dictionaries. The dictionary store objects and the synonyms while using the ontology tree and the document vector of each web document has been prepared. Ontology tree is created to show the relationship among the objects or terms in a web document. From ontology tree, the number of occurrence of different terms in a particular web document has been calculated.

The algorithm for creating ontology tree for each web document is described below.

1) For each newly arrived web document x[i]; where i = 1,2,…,p.
2) Take a user defined domain name as Root.
3) Repeat from i = 1.
4) Take a web document present in x[i] and sort the words present in it alphabetically with hash indices and initialize the words as word(j); where j = 1,2,…,n. This helps to maintain the hash indices.
5) Mark x[i] as the child of the Root.
6) Now search each word(j) in the concept dictionary. If the word is found, make it a child of the corresponding concept found in the concept dictionary and mark the concept as a child of x[i].

7. a) If the word(j) present in the synonym or the antonym dictionary, make it a child of the corresponding object represented by object tag and search for it in the concept dictionary to find its corresponding concept tag, which is treated as parent of the object.

7. b) Else, search the sub-tree rooted in document x[i] in ontology tree for its existence. If found, make the corresponding concept as parent of that object, else create a new concept node and make it a child of x[i] while the object becomes a child of that concept as shown in Figure 5.

When under one document x[i] two or more concept will be there in ontology tree, it means the document is of mixture concept; otherwise the web document is of pure concept.

The Ontology Tree creation for two documents are shown in Figure 5, where the numbers written on the objects or terms in the ontology tree signifies the number of occurrences of that particular object or term in that particular web document. The process of storing occurrences of objects described in step 5.

Step 5: Preparation of document Vector

For each document, a document vector has been created with the help of ontology tree consisting of number of occurrences of different objects of different concepts of the concept dictionary. From the ontology tree the

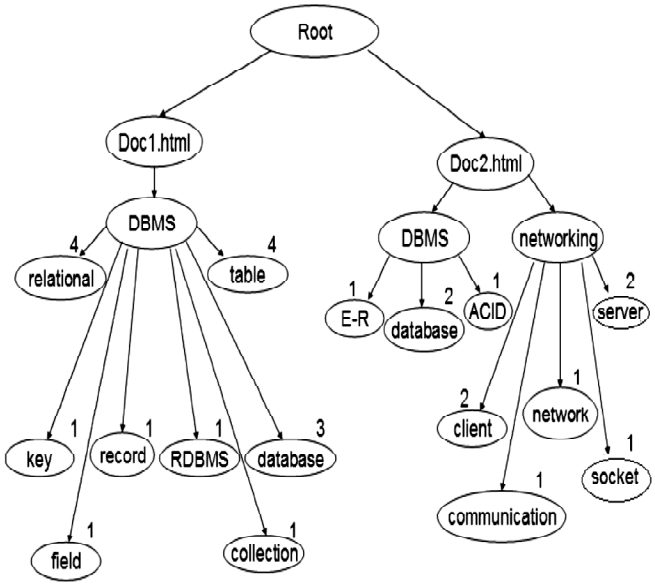

Figure 5: Ontology based tree for Doc1.html and Doc2.html

occurrence of only the objects present in the web document is calculated to create document vector. The occurrence of synonym object node in ontology tree will increase the number of occurrences of its parent object occurrences. It means, suppose a term MAN is found in a web document, but in the concept dictionary the object is LAN. Then MAN will increase the occurrence of the object LAN in document vector, as MAN is representing a sub object node of parent object node LAN. The concept of document vector is shown below in figure 6.

	$Term_1$	$Term_2$	$Term_3$	$Term_4$		$Term_n$
Doc	N_1	N_2	N_3	N_4	.....	N_n

Figure 6: Document vector

Here $N_1, N_2 \ldots N_n$ are the total number of occurrences of objects of concept dictionary.

Step 6: Preparation of concept Vector

Now from the document vector, a concept vector of a newly arrived web document is created, which is built from total objects related to different concepts, is shown below in figure 7.

Here T_1 is total terms or objects of networking concept whereas T_2 is total terms or objects of DBMS concept whereas T_3 is total terms or objects of operating system concept.

Doc	T1	T2	T3
	Networking	DBMS	Operating System

Figure 7: Concept vector

Step 7: Calculating concept index of a new arrived web

Here, a new approach for evaluating the internal concept of the newly arrived web document has been discussed. An index D_i of new web document DOC_{new} has been calculated by the following method. In this paper, clustering is done on the basis of 3 concepts - networking, DBMS, Operating System. As per figure 7, it is seen that T1, T2 and T3 are the total terms or objects related to different concepts in a document. On the basis of some conditions related to T1, T2 and T3, the concept index of the newly arrived document has been mentioned below in Table 1.

Table 1: Concept Index generation

Condition	*Index*
1. T1 > T2 > T3 = 1	8. T1 = T2 > T3 = 8
2. T1 > T3 > T2 = 2	9. T1 > T2 = T3 = 9
3. T2 > T1 > T3 = 3	10. T2 > T1 = T3 = 10
4. T2 > T3 > T1 = 4	11. T2 = T3 > T1 = 11
5. T3 > T1 > T2 = 5	12. T3 > T1 = T2 = 12
6. T3 > T2 > T1 = 6	13. T3 = T1 > T2 = 13
7. T1 = T2 = T3 7	

Step 8: Process of dynamic web document clustering

Condition 1: if no cluster exists then DOC_{new} will create a new cluster.

Suppose cluster set {NULL}, the DOC_{new} will create a new cluster set like $\{DOC_{new}\}$.

Condition 2: if some clusters already exist, then do the following:

Suppose in a cluster document doc1 and document doc5 are present. Now these document's concept vectors can create cluster concept matrix (CCM) of that cluster as below:

	Networking	DBMS	Operating System
Doc1	T_1	T_2	T_3
Doc5	T_1	T_2	T_3

Now, for calculating the concept Index of a particular cluster, the mean values of different concepts have been calculated using the procedure below for n number of documents in a cluster:

for j=1 to 3 //because of 3 concepts

$$T_{j\,avg} = \sum_{i=1}^{n} a_{ij};$$

where a_{ij} is the element of the CCM for different concepts.

So, from the procedure above, the mean of each concept suppose T1avg, T2avg, T3avg will be calculated to get the cluster centre.

Now, the cluster centre is like a concept vector as described in figure 7 for this existing cluster. The cluster centre vector (CCV) is shown below in figure 8.

T_{1avg}	T_{2avg}	T_{3avg}

Figure 8: Cluster Centre Vector

Now, from this CCV, the cluster index Ci can also be created by the procedure mentioned in step 5 and step 6 for a particular cluster.

I) Next, index of each existing cluster is decided by the use of CCV for each cluster. Suppose, the cluster index vector (CIV) containing all the indices of different clusters is shown below:

CIV = { C_1, C_2, C_3 . . . . . C_n}

II) Now the index of new document D_i is compared with all cluster indices exist in cluster index vector.

III) Now DOC_{new} will be inserted in the cluster with same index from CIV.

IV) Now the above process will be continued for all new arrived documents.

RESULT ANALYSIS

For experimental purpose, the 120 tutorial web documents collected from tutorial site http://www.tutorialspoint.com and http://www.indiabix.com/technical/interview-questions-and-answers for three types of concepts networking, DBMS, Operating System and some are of mixture concept. The effectiveness of the algorithm proposed here is justified by using standard cluster quality measure like F-Measure and purity.

F-measure: F-measure combines the precisions and recall from information retrieval process. The precisions and recall of a specific cluster for a given class are calculated. More specifically F-measures for cluster j and cluster i are calculated as follows;

$$\text{Recall (i,j)} = \frac{n_{ij}}{n_i}$$

$$\text{Precision (i,j)} = \frac{n_{ij}}{n_j}$$

$$\text{F (i,j)} = \frac{(2*\text{Re}call(i,j)*\text{Pr}ecision(i,j))}{\text{Pr}ecision(i,j)+\text{Re}call(i,j)}$$

Where n_{ij} is the number of members of the class i in cluster j, n_j is the number of members of cluster j and n_i is the number of members of class i. For each class, only the cluster with highest F-measure is selected. Finally, the overall F-measure of a clustering solution is weighted by the size of each cluster:

$$\text{F(S)} = \frac{1}{n}\sum_{j=1}^{n}\frac{n_j}{\max(F(i,j))}$$

In general, larger value of F-measure means better clustering solution.

Purity: It is one of the cluster validity measures. Let there be k clusters of the dataset n and size of cluster C_i be $|C_i|$. Let $|C_i|_{class=j}$ is number of items of class j given to cluster i. Then purity of cluster is given by the following:

$$(C_i) = \frac{1}{|C_i|}(|C_i|_{class=j}).$$

The overall purity of clustering result can be expressed as a sum of all cluster purities. *Purity* = $\sum_{j=1}^{k}\frac{C_i}{n}$ purity (C_i)

In general, larger value of purity means better clustering solution.

An experiment based on 120 tutorial web documents has been done for dynamic web document clustering. The documents have been inserted in the existing clusters or they have created new cluster. Here, two comparisons have been done between proposed algorithm and the procedure suggested by Shehata, S. (2010) and procedure CCFICA suggested by Jayabharathy *et al.* (2012). The algorithm proposed in this paper has the advantages over the above said algorithms in respect of less time complexity because less time is required for searching the objects or terms of concept dictionary in the web documents using ontology tree and for the creation of document vector as well as concept vector. The concept vector is responsible for creating index of a newly arrived document. This is possible with less time only for using hash index organization of documents and dictionary. As only the objects or terms present in the ontology tree of a particular web document have been searched in concept dictionary, so less time requirement for

creation document vector. The proposed algorithm performs better than the existing CBA algorithm proposed by Shehata, S. (2010) because CBA takes only the Synonyms / Hyponyms in the dataset and but in this paper the internal concepts of the web documents have been considered with lesser error. In compare to CCFICA algorithm, this paper suggests a clear concept of dynamic and efficient method of web document clustering with less time complexity and more accurate output with clearer and meaningful concept analysis. Table 2 presents the results of F-measure and Purity for CBA, CCFICA and New Proposed Algorithm on the experimental datasets.

Table 2: F-measure and Purity Results

Algorithm	*F-measure*	*Purity*
CBA	0.523	0.762
CCFICA	0.582	0.801
New Proposed Algorithm	0.625	0.873

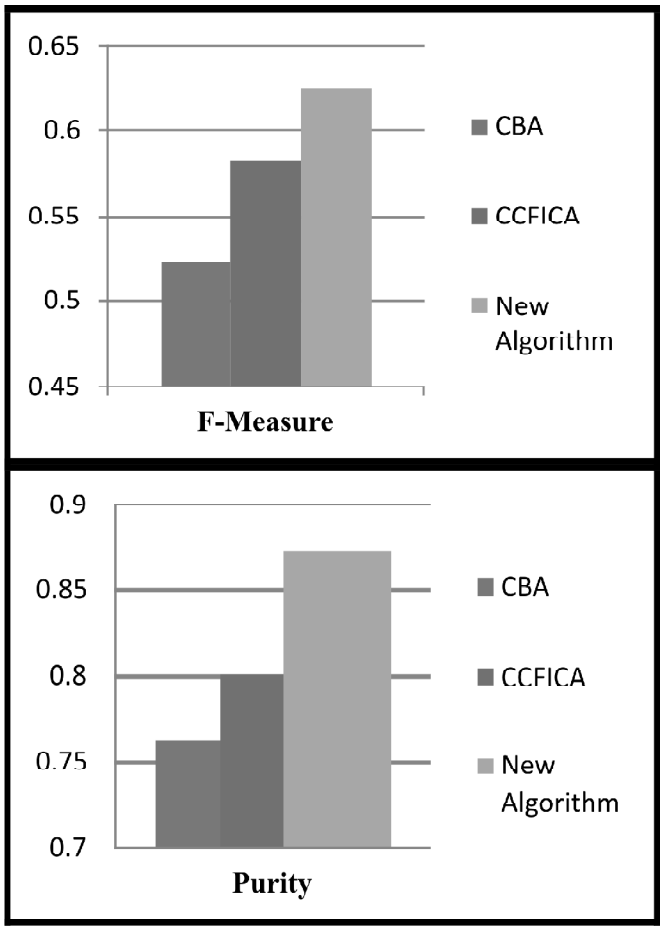

Figure 9: Comparison of F-measure and Purity Values

Figure 9 shows that the new proposed algorithm is more effective and more accurate over the existing algorithms in respect of F-Measure and purity.

CONCLUDING REMARKS

In this paper, an efficient method of dynamic web document clustering has been proposed where time complexity of the searching algorithm for objects in a dictionary as well as in web documents is less because of maintaining hash indexing and different tags used in the dictionaries. On the other hand, analyzing the concept of a new arrived document as well as analyzing the concept of the existing clusters by the use of concept similarity index becomes much more efficient and less time consuming. The procedure of evaluating the index of a newly arrived document and that of a cluster is very much suitable for considering the entire concept as well as the sub concepts present in new web document as well as existing clusters. In CCFICA algorithm the similarity analysis is not clearly described, whereas in the algorithm proposed in this paper the concept analysis is clearly described. The table 2 shows the effectiveness of the new proposed algorithm, where the overall performance in respect of internal concept analysis of web document and time complexity of the new proposed algorithm are better than existing CBA algorithm and CCFICA algorithm.

Biographical Notes

Soumen Swarnakar is assistant professor in the department of Information Technology at Netaji Subhash Engineering College, Kolkata, India. He received M.E. degree from Bengal Engineering & Science University, Shibpur, Howrah, India. His area of interest is Data Mining, Web Services.

Sreya Dutta, Sayni Sarkar, Kausani Chakroborty, Saheli Kar are B.Tech final year students in the department of Information Technology, Netaji Subhash Engineering College, Kolkata, India.

References

[1] Han, J. and Kamber, M. (2001), Data Mining: Concepts and Techniques, 2nd ed., Morgan Kaufmann Publishers, CA, USA.

[2] Jayabharathy and Kanmani. (2012), "Correlated concept based dynamic document clustering algorithms for newsgroups and scientific literature". Decision Analytics Journal.

[3] Nagaraj, R. and Thiagarasu , V. (2014), "Correlation Similarity Measure based Document Clustering with Directed Ridge Regression". Indian Journal of Science and Technology. Vol. 7, issue5, pp. 692-697.

[4] Nageswara Rao, B., Keerthi, P., SreeRamya, V.T., Santhosh Kumar, S. and Monish, T(2014), "Implementation on Document Clustering using Correlation Preserving Indexing". International Journal of Computer Science and Information Technologies, Vol. 5 (1), pp. 774-777.

[5] Oikonomakou, Nora. and Vazirgiannis, Michalis. (2005), "A Review of Web Document Clustering Approaches". Data Mining and Knowledge Discovery Handbook, Sringer US. pp. 921-943.

[6] Peng, Jing. Yang, Dong-qing. Wang, Jian-wei. Wu, Meng-qing. Wang, Jun-gang.(2007), "A Clustering Algorithm for Short Documents Based On Concept Similarity". IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, PacRim 2007, pp. 42 – 45.

[7] Scheker, Adam., Last, Mark., Bunke, Horst. And Kandel, Abraham. (2004), "Comparison of Algorithms for Web Document Clustering Using Graph Representations of Data". Proceedings-Joint IAPR International Workshops, SSPR 2004 and SPR 2004, Lisbon, Portugal, pp. 190-197.

[8] Shehata, S. (2010), "An efficient concept-based mining model for enhancing text clustering". Journal of Knowledge and Data Engineering, Vol. 22, No. 10, pp. 1360–1371.

[9] Steinbach, M., Karypis, G., & Kumar, V. (2000), "A Comparison of Document Clustering Techniques". International Conference on Data Mining: Knowledge Discovery and Data Mining (KDD) Workshop on Text Mining, pp. 1–2.

[10] Wang, Xufei. Tang, Jiliang. and Liu, Huan . (2011), "document clustering via matrix representation". Proceedings - IEEE International Conference on Data Mining, ICDM, pp. 804-813.

[11] Yan, Yang. , Chen, Lihui. And Tjhi, William-Chandra. (2012), "Fuzzy semi-supervised co-clustering for text documents". Fuzzy Sets and Systems, Vol.215, pp. 74–89.

[12] Zheng, Xiao-Shen., He, Pi-lian., Tian, Mei. and Yuan, Fu-yong. (2003), "Algorithm of documents clustering based on minimum spanning tree". IEEE Second International Conference on Machine Learning and Cybernetics, Vol. 1, China, pp- 199 – 203.