International Journal of Computer Science and Business Informatics

IJCSBI.ORG

A Hybrid Algorithm for Improvement of XML Documents Clustering

Somayeh Ghazanfari

Department of Computer Engineering, Islamic Azad University,
Science and Reasearch Campus Lorestan , Iran

Hassan Naderi

Department of Computer Engineering, Iran University of Science and Technology,
Resalat St., Tehran, Iran

ABSTRACT

As Extensible markup language (XML) documents are now widely used in the Web World, improving the speed and accuracy of search engines based on these documents is important. Clustering is a way that can be effective in improving the speed of the search engine. Clustering of XML documents can be divided into pair wise and incremental algorithms. The main challenge in the class of incremental algorithms such as Level Structure (XCLS), XCLS+ and XCLS++ is that the order of input XML documents influences the clustering. In this paper, the sensitivity of incremental XML clustering algorithms is introduced by a representative algorithm i.e. XCLS+. A typical solution to this problem has been proposed which includes two interleaved phases: online and semi-offline. Experimental results show that the proposed algorithm has a higher speed with a relatively higher precision for large number of documents compared to previous incremental algorithms such as XCLS+.

Keywords

Incremental algorithms, XML clustering, XCLS+, Input of documents.

1. INTRODUCTION

The popularity use of web and internet causes a large amount of data and information. The growth of stored data requires automatic tools to allow the transformation of large amounts of data into information and knowledge intelligently. Data mining is proposed as a solution for this issue [1] and clustering is an important technique applied in data mining. A cluster refers to a set of data which have most similarity to each other (inter-class) and are less similar to other clusters (intra cluster). Clustering can be applied on various types of data such as images, numbers, documents and texts. Nowadays XML becomes the standard for data transmission and development [2] and most internet data is in semi-structured forms such as

XML documents [3]. This is because of the simplicity, expandability, easy access and openness of XML. XML is one of the data types that clustering can be performed on it.

One of the problems in incremental algorithms is that, the order of input of documents, affects the clustering. In this paper, a new algorithm called WXCLS + is introduced to reduce the sensitivity in the order of input of documents in incremental clustering algorithms. The new algorithm clusters with a combination of online and hierarchical clustering which is an offline method. With this method, a cluster prevents enlargement and enhance the accuracy of clustering.

This paper is organized as follows. Section 2 discusses related works on XML document clustering algorithms especially incremental ones. Section 3 introduces the problem of sensitivity of incremental clustering algorithms to the order of input of documents. XCLS+ algorithm has been used in this paper as a sample to show its problem and to compare with our proposed method. The algorithm for overcoming these hardships has been proposed in Section 4. The simulation results and algorithm evaluations via experiments are presented in section 5, and finally sections 6, 7 conclude the paper and suggest the future works of the paper, respectively.

2. RELATED WORK

The clustering algorithms of XML documents can be categorized into pair wise and incremental approaches. In pair wise approach, the clustering algorithms are supposed to posses all documents at first. Some of these methods might also investigate each document repeatedly. On the other hand, possessing only one document at a time in the incremental approach; therefore, it must investigate and cluster each document only once. The main goal of these approaches is higher speed in clustering while maintaining acceptable accuracy. Using a global criterion for computing similarity is a considerable point in incremental methods. In order to decrease the processing time, incremental algorithms reduce each cluster by just maintaining a document representative. A cluster representative is an aggregated document which combines cluster documents in a single document. To be able to cluster a set of documents, it's necessary to have a similarity calculation method. Document's similarity with a cluster representative is an appropriate measure for determine its cluster. Methods to determine the similarity between one XML document and a representative cluster are generally divided into three categories: content-based [10], structure-based [2] [5] [11] and combination of mentioned methods [1] [3] [4] [8] [9]. For example, XCLS, XCLS+, and XCLS++

algorithms are all incremental algorithms that consider the structure of documents [5,6]. XCLS performs clustering well for the heterogeneous documents, but does not consider the node relations in the tree structure; therefore, it is not proper for homogeneous documents. The XCLS+ algorithm is introduced after XCLS, which have more information in its level structure compare to XCLS method, also as well as to the elements name, contains information about their parents. The XCLS+ similarity criteria are performed based on parent-child relationship [7]. The XCLS++ algorithm has improved the similarity criteria of the XCLS+ algorithm By considering father-child relations. Despite all attempts which have been done to improve clustering in XCLS, XCLS+, XCLS++ algorithms, they still suffer from the problem of sensitivity to the order of input documents. It means, different results are observed in clustering by changing the input documents order. This problem occurs when very similar documents are entered after each other (homogeneous documents). XCLS+ has been selected in this research as an example to show the mentioned problem and to be evaluated with our proposed method.

3. XCLS+ ALGORITHM

In XCLS+ each XML document and cluster representative is modeled by a level structure object. The Level structure stores element's parent as well as the element's name. The new input document is compared with updated Level structure of the clusters. This new document will be merged with a most similar cluster representative (Fig. 1) [5].

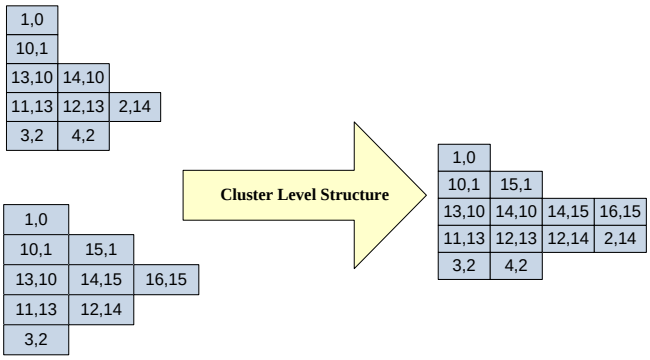

Fig. 1: Cluster Level structure merging in XCLS+ method

The formula to be used for calculating the similarity between a XML document and cluster representative is as follow:

$$\text{LevelSim}_{1->2} = \frac{0.5 \times \sum_{i=0}^{L-1} (CN_1^i + CP_1^i) \times r^{L-i-1} + 0.5 \times \sum_{j=0}^{L-1} (CN_2^j + CP_2^j) \times r^{L-j-1}}{\left(\sum_{k=0}^{L-1} N^k \times (r)^{L-k-1} \right) + 0.5 \times \left(\sum_{i=0}^{L-1} CP_1^i \times r^{L-i-1} + \sum_{j=0}^{L-1} CP_2^j \times r^{L-j-1} \right)}$$

(1)

The parameters used this formula are:

- CN_1^i Sum of occurrences of every common element in the level i of the object 1 .
- CN_2^j Sum of occurrences of every common element in the level j of the object 2 .
- CP_1^i Number of occurrences of all common elements in level i of the object 1 which have the same parent.
- CP_2^j Number of occurrences of all common elements in level j of the object 2 which have the same parent.
- N^k Number of elements in level k of the document.
- R Base Weight: the increasing factor of weight. This is usually larger than 1 to indicate that the higher level elements have more importance than the lower level elements.
- L Number of levels in the document.

In equation (1), CP indicates the Number of all common elements which have the same parent while it is clearer in homogeneous XML document. Instead of using number tags for elements, their own names are used in order to perform a full search in XCLS+ algorithm. The Fig. 2 indicates a tabular view of a document including element name (Tag Name), Parent name (Parent), and level number (Level).[5]


```
<?xml version="1.0" encoding="ISO-8859-1"?>
<DOCTYPE W4F_DOC SYSTEM "actors.dtd">
<W4F_DOC>
  <Actor>
    <Name>
      <FirstName>  Donnie      </FirstName>
      <LastName>   Wahlberg    </LastName>
    </Name>
    <Filmography>
      <Movie>
        <Tiltle>  Altoona Riding Club  </Tiltle>
        <Year>  2000    </Year>
      </Movie>
      <Movie>
        <Tiltle>  Purgatory       </Tiltle>
        <Year>  1999       </Year>
      </Movie>
    </Filmography>
  </Actor>
</W4F_DOC>
```

Tag Name	Level	Parent
W4F_DOC	0	
Actor	1	W4F_DOC
Name	2	Actor
FirstName	3	Name
LastName	3	Name
Filmography	2	Actor
Movie	3	Filmography
Tiltle	4	Movie
Year	4	Movie

Fig. 2: Tabular view of a XML document suitable for XCLS+ clustering

The steps of matching of two objects in XCLS+ method are as follows:

1) First, the Level structures of both objects must be turned into tabular presentation. The tables must be arranged based on element names.
2) Then, start with searching for common elements in the first level of both tables. If at least one common element is found, mark the number of common elements with the level number in object 1 (CN_1^0) and the number of common elements with the level number in object 2 (CN_2^0), then go to step 3 otherwise, go to step 4 .
3) Move both objects to the next level tables (level i++, level j++) and search for common elements in these new levels; if at least one common element is found, mark the number of common elements with the level number in object 1 (CN_1^i)) and the number of common elements with the level number in object 2 (CN_2^j), then go to step 3. Otherwise, go to step 4.
4) the element names are compared, and as the element names are sorted, the change of level only occurs in a table row where the element's name is smaller. Because, in the next table row with smaller element name, the possibility of finding common elements exists, but the contrary is impossible.
5) Matching continues until one table reaches its final row.

This structural matching for two objects has the advantage of finding all common elements between both objects. To find common elements with the same parents we do the same as explained in finding common elements.

With all advantages of XCLS+ against XCLS method, it has some problems which are introduced below. [5]

3.1 First problem of XCLS+ and its solution

According to Equation (1) LevelSim is a value between 0 and 1; 0 indicates completely different objects and 1 indicates homogenous objects. LevelSim is not symmetric, meaning that LevelSim1→2 is different with LevelSim2→1. Asymmetry is problematic when the documents are homogenous, and even in some cases the similarity will be more than 1. Fig. 3 shows the structure of two XML homogenous documents named Movie1 and Movie2.

```
<? Xml version="1.0" encoding="ISO-8859-1"?>
  <W4F_DOC>
    <Movie>
      <Title> Nosferatu,eine Symponie des
Grauens</Title>
      <Year> 1922 </Year>
      <Directed_By>
        <Director> F.W.Muranau </Director>
      </Directed_By>
      <Generes>
        <Genere> Horror </Genere>
        <Genere> (more) </Genere>
      </Generes>
      <Cast>
        <Actor>
          <FirstName> Gustav </FirstName>
          <LastName> Botz </LastName>
        </Actor>
      </Cast>
    </Movie>
  </W4F_DOC>
```

```
<? Xml version="1.0" encoding="ISO-8859-1"?>
  <W4F_DOC>
    <Movie>
      <Title> Smultron stallet </Title>
      <Year> 1957 </Year>
    </Movie>
  </W4F_DOC>
```

Fig. 3: Two sample XML documents (Movie1,Movie2)

Using XCLS+ formula we have:

$$\text{LevelSim1} \rightarrow 2 = \frac{0.5(2*2^4+2*2^3+4*2^2+0*2^1+0*2^0)+0.5(2*2^2+2*2^1+4*2^0)}{(1*2^4+1*2^3+5*2^2+3*2^1+2*2^0)+0.5((1*2^4+1*2^3+2*2^2+0+0)+(1*2^2+1*2^1+2*2^0))} = 0.555$$

$$\text{LevelSim } 2 \rightarrow 1 = \frac{0.5(2*2^4+2*2^3+4*2^2+0*2^1+0*2^0)+0.5(2*2^2+2*2^1+4*2^0)}{(1*2^2+1*2^1+2*2^0)+0.5((1*2^4+1*2^3+2*2^2+0+0)+(1*2^2+1*2^1+2*2^0))} = 1.428$$

This significant difference between two objects Movie1 and Movie2 is due to N^k variable in the denominator of the formula. With such variable, the

number of input document nodes will be important. For example, in homogenous documents due to existence of many common nodes, the numerator of the fraction is large. Now a) if the input document has fewer amounts of common nodes, the denominator of function is small and in result the similarity is high. b) If the input document has more amounts of nodes, the denominator of function is large and in result the similarity is decreases dramatically.

3.2 Our solution for this problem

To solve this problem, the similarity formula can be redefined as follows:

$$\text{LevelSim}_{1->2} = \frac{\sum_{i=0}^{L-1}(CN_1^i + CP_1^i)\times r^{L-i-1} + \sum_{j=0}^{L-1}(CN_1^j + CP_1^j)\times r^{L-j-1}}{\sum_{k=0}^{L-1} N^k \times (r)^{L-k-1} + \sum_{k=0}^{L-1} M^k \times (r)^{L-k-1} + \sum_{i=0}^{L-1} CP_1^i \times r^{L-i-1} + \sum_{j=0}^{L-1} CP_1^j \times r^{L-j-1}} \quad (2)$$

In this formal a new variable M^k is defined. M is the number of nodes in k-th level of comparison cluster. Using this formula the similarity of documents of Fig 3 is as follow:

$$\text{LevelSim}1 \rightarrow 2 = \text{LevelSim}2 \rightarrow 1 = 0.8$$

So, not only formula 2 is symmetric but also its result is more acceptable than that the result of formula 1.

3.3 Second problem of XCLS+

After comparing two documents and determining their related similarity value, if any, XCLS+ algorithm merges them. In the cases that the number of documents is high, changing the sequential order of input documents, affect the results of clustering algorithm. So the algorithm encounters difficulty to appropriately distinguish the right cluster for the input document. In our solution to this problem, we will first pre-cluster the input document into a more coherent cluster. It means that for clustering a document, its similarity to the cluster representative has to be sufficiently decisive; otherwise the new document will create a new cluster. The result of this decision will be a lot of small clusters in the first phase of our approach. In the second phase, these small rigid clusters will be merged by an offline algorithm to create a list of final appropriate clusters. Using this hybrid approach (combination of an online and an offline clustering algorithm) would permit gaining the speed of online algorithms as well as the precision of offline algorithms. This approach is called **WXCLS+** and will be described in detail.

Input:

O1,O2: Two Novel Level Structure represented **as Table** which **Ordered by** TagName, Level

OL1 : the number of row of O1 in $\{OL1_1, OL1_2, \ldots, OL1_w\}$

OL2 : the number of row of O2 in $\{OL2_1, OL2_2, \ldots, OL2_z$

Output:

CN1[1..w] as number

CN1[1..z] as number

Method:

(1) **Repeat**
(2) Compare each row O1 with each row j of O2
(3) If (O1.TagName = O2.TagName) {
CN1[$OL1_i$] ++ ;
CN2[$OL2_j$]++;
Go to the next rows of both object O1 ,O2;
}
(4) If (O1.TagName > O2.TagName)
Go to the next row of just O2
(5) Else
Go to the next row of just O1
(5) Until all rows of both objects checked
(6) PHASE 1: ONLINE CLUSTERING
(7) If (bunchSize<bunchCapacity){
(8) Add input document to the current bunch using XCLS+
(9) }
(10) Else {
(11) Create a new bunch and add input document to the new bunch
(12) }

PHASE 2: OFFLINE CLUSTERING

(13) For each pair of clusters in all bunches
{
(14) Calculate the similarity between two clusters
(15) If (similarity<=threshold) {
(16) Merge this pair of clusters}

Fig. 4: WXCLS+ a hybrid clustering algorithm to overcome the problem of order of input documents

4. COMBINATION OF ONLINE WITH OFFLINE CLUSTERING ALGORITHMS

WXCLS+ calculates the similarity of documents using our proposed equation (2). This new formula uses a level structure to obtain the similarity of XML documents. However, the main difference between the new technique and previously suggested incremental algorithms such as XCLS+ is in the clustering process. Clustering process for the prior methods is done completely in online method, however, simultaneous offline and online clustering process was utilized for this new method to overcome the problem of order of input documents. Fig 4 shows the proposed algorithm. According to this algorithm, new clustering is performed in two phases: online (incremental) phase and offline (hierarchical) phase.

Online phase

A threshold variable named BunchSize is defined for the maximum size of incrementally created clusters. So, if we consider N as the number of whole documents, the number of categories will be at least k= N/BunchSize. The major difference in the new way is creating smaller but rigid clusters which are considered for comparison and documents clustering.

Offline phase

After termination of documents, there exists a number of sufficiently small and rigid clusters. In the second phase, these clusters are combined using a hierarchical clustering algorithm. In other words, when the traffic load is low and the number of clusters is higher than a determined value, clustering is performed in offline mode using a hierarchical clustering algorithm. Using this hybrid approach (combination of an online and an offline clustering algorithm) would permit gaining the speed of online algorithms as well as the precision of offline algorithms. Moreover, in previous methods by increasing the number of documents and creating bigger clusters, comparison between a document and a cluster representative is so time consuming which will reduce the speed of comparison. It must be mentioned that this algorithms is sensible to the value of BunchSize. If its value is very small, the overhead of the program will be increased. Instead, if the value is very large the the algorithm as previous algorithm suffers from the problem of enlargement of cluster representative. Fig. 5 shows graphically different steps of proposed clustering algorithm.

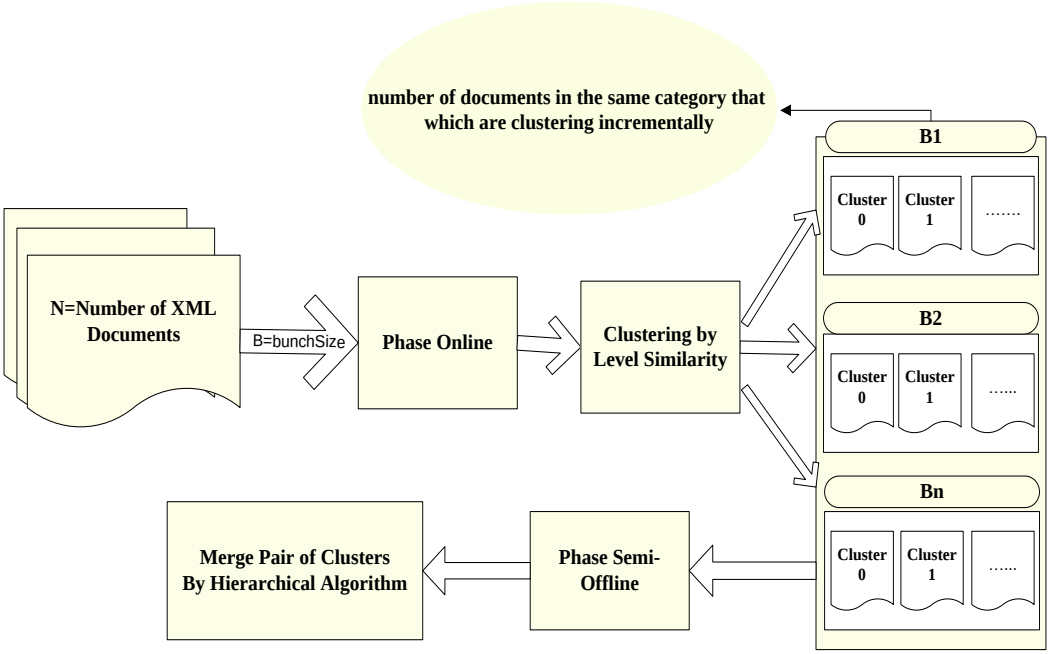

Fig. 5: Clustering WXCLS+ method

5. Evaluation of the proposed method

Both XCLS+ and WXCLS+ methods are implemented by Microsoft visual studio2010 using the programming language C #. Three external criteria of Entropy, Purity and Fscore [3], [17] are used to compare these two methods. The evaluation criteria were performed in the same conditions on a data set. Two data sets are used to evaluate the performance of WXCLS+ against XCLS+ including both homogenous documents (single type DTD)[15] and heterogeneous documents (multi-type DTD) [16]. Results of both sets are examined and shown separately. The heterogeneous documents set consists of 700 XML documents, while homogenous documents contain 120 department documents consisting of four Sub_DTDs.

At first, both clustering approaches are applied on the set of heterogeneous data consisting of 700 documents. Tables 1, 2, 3 and 4, show the clustering results for two methods using different threshold values With the Same Order of Input Documents and with different Order of Input Documents in XCLS+ & WXCLS+.

Table 1: The Results on heterogeneous documents with Same Order of Input Documents

AlGORITHM	THRESHOLD		ENTROPY		PURITY		FSCORE	
	WXCLS+	XCLS+	WXCLS+	XCLS+	WXCLS+	XCLS+	WXCLS+	XCLS+
	0.7	0.7	0.05	0.03	0.9	0.9	0.9	0.9
	0.8	0.8	0	0.01	1	0.9	0.9	0.9
	0.9	0.9	0	0.01	1	0.9	0.9	0.9

Table 2: The Results of algorithm XCLS+ on heterogeneous documents with different Order of Input Documents

AlGORITHM	THRESHOLD	ENTROPY Changing the order of documents for three time			PURITY Changing the order of documents for three time			FSCORE Changing the order of documents for three time		
XCLS+	0.7	0	0.04	0.02	1	0.92	0.95	0.99	0.93	0.96
	0.8	0	0	0	1	1	1	0.99	0.99	0.99
	0.9	0	0	0	1	1	1	0.98	0.97	0.98

Table3: The Results of algorithm WXCLS+ on heterogeneous documents with different Order of Input Documents (Bunch Size = 50)

AlGORITHM	THRESHOLD	ENTROPY Changing the order of documents for three time			PURITY Changing the order of documents for three time			FSCORE Changing the order of documents for three time		
WXCLS+	0.7	0.05	0.05	0.05	0.92	0.92	0.92	0.93	0.93	0.93
	0.8	0	0.01	0	1	0.97	1	0.99	0.97	0.99
	0.9	0	0	0.01	1	1	0.97	0.99	0.99	0.99

Table 4: The Results of algorithm WXCLS+ on heterogeneous documents with different Order of Input Documents (Bunch Size = 100)

AlGORITHM	THRESHOLD	ENTROPY Changing the order of documents for three time			PURITY Changing the order of documents for three time			FSCORE Changing the order of documents for three time		
WXCLS+	0.7	0.06	0.05	0.05	0.9	0.92	0.92	0.91	0.93	0.93
	0.8	0	0	0	1	1	1	0.99	0.99	0.99
	0.9	0	0	0	1	1	1	0.99	0.99	0.99

In new method we were looking for an approach which clustering process is not affected by altering the order of input documents. This goal was achieved based on the results is these four the tables. Another objective of WXCLS+ method was to improve clustering results compared to XCLS+ method. Whereas in some section of these four tables, the XCLS+ method is better than our method. The reason behind this fact is that because XCLS+ and WXCLS+ algorithms consider parent-child relationship in comparing documents, they are more effective in evaluation of homogeneous documents. The methods WXCLS+ and XCLS+ will be evaluated for homogenous documents and it will be seen that WXCLS+ method compared to XCLS+ has considerable improvement.

To evaluate the results of the XCLS+ and WXCLS+ methods on homogeneous documents, we have used the DTD of the department to create 4 sub DTDs. We created a total number of 80 homogeneous XML documents. Nodes of faculty, staff and grad student were eliminated from Sub_DTD1. While, nodes of undergrad student, faculty and staff were removed from Sub_DTD2. In the case of Sub_DTD3, nodes of undergrad student, staff and grad student were omitted, whereas, undergrad student, faculty and grad student were discarded in Sub_DTD4.

To evaluate the homogeneous documents, all of the sub-DTDs are put into one class. then the documents' position are changed to see the evaluation criteria in different threshold values at the output. Fscore criterion is used here for the evaluation of both XCLS+ and WXCLS+ methods. Tables 5 and 6 show the evaluation results by several changes of the order of the input documents for the methods XCLS+ and WXCLS+. To have a better evaluation, several states are considered for documents formerly and give them as inputs into the program to obtain the evaluation results in identical conditions.

Table 5: The Results of the evaluation on homogeneous documents by method XCLS+

AlGORITHM	**THRESHOLD**	**FScore** (first)	**FScore** (Second)	**FScore** (Third)	**FScore** (Fourth)
XCLS+	0.8	0.84	0.9	0.89	0.98
	0.87	0.46	0.9	0.89	0.98

Table 6: The Results of the evaluation on homogeneous documents by method WXCLS +(BunchSize=25)

AlGORITHM	THRESHOLD	FScore (first)	FScore (Second)	FScore (Third)	FScore (Fourth)
WXCLS+	0.8	1	1	1	1
	0.87	1	1	1	1

Tables 5 and 6 show that the proposed new method has better results in comparison to XCLS+ method and performs exact clustering with the threshold values 0.8 and 0.87.

6. CONCLUSION AND FUTURE WORK

The incremental algorithms like XCLS and XCLS+ perform clustering process with an acceptable speed. However, a careful study of the XCLS+ shows two major problems: (1) asymmetry in the computation of structural similarity between two documents based on defined similarity formula, 2) because of the incremental nature of the algorithm, with increasing the number of documents, clusters are grown and the quality of clustering process is decreased. To give a solution for these problems, two proposals are offered: (1) by defining a new variable, asymmetry problem for calculation of similarity between two documents (document with clustering) has been resolved; (2) by combining two offline and online clustering algorithm, we avoid the enlargement of cluster's representative which affects the quality and speed of clustering process. The advantage of this new approach is that accuracy and speed of clustering process are significantly improved.

In this paper, the three criteria Purity, Entropy and Fscore were used for evaluation. But these criteria in Homogeneous documents have some problems which in the future work new evaluation criteria for the evaluation algorithms can be defined. Another issue that can be addressed is the similarity formula between the documents. If a formula is provided that the variables are less, will significantly increase the efficiency of algorithm.The final proposal is that the other offline algorithms will be combined with online method in order to cluster XML documents.

REFERENCES

[1] Jiawei, H. and Kamber, M., 2001. Data mining: concepts and techniques. San Francisco, CA, itd: Morgan Kaufmann, Vol. 5.

[2] Bray, T. Paoli, J. Sperberg-McQueen, C.M. Maler, E. and Yergeau, F., 2004. Extensible markup language (xml) 1.0.

[3] Nayak, R., 2008. Fast and effective clustering of XML data using structural information, Knowledge and Information Systems, Vol. 14, No. 2, pp. 197–215.

[4] Nayak, R. and Xu, S., 2006. XCLS: a fast and effective clustering algorithm for heterogeneous XML documents, In Advances in Knowledge Discovery and Data Mining Springer Berlin Heidelberg.

[5] Alishahi, M. Ravakhah, M. Shakeriaski, B. and Naghibzade, M., 2009. XML document clustering based on common tag names anywhere in the structure, In Computer.

[6] Naghibzadeh, M., 2010. Tag Name Structure-based Clustering of XML Documents, International Journal of Computer and Electrical Engineering (IJCEE), No. 2.

[7] Qaramaleki, A. K. E. and Naderi, H., 2013. A New Online XML Document Clustering Based on XCLS++, International Journal of Computer Science and Business Informatics,Vol. 2, No. 1.

[8] Nierman, A. and Jagadish, H. V., 2002. Evaluating Structural Similarity in XML Documents, In WebDB, Vol. 2, pp. 61-66.

[9] Peng, J. Dong, Q. and Yang, S., 2008. Similarity in Chinese text processing, A New Similarity competing method based on concept, series F: Information science, Vol. 51, No. 9, pp. 1212-1230.

[10] Ghosh, S. and Mitra, P., 2008. Combining content and structure similarity for XML document classification using composite SVM kernels, In ICPR, pp. 1-4.

[11] Choi, I. Moon, B. and Kim, H. J., 2007. A clustering method based on path similarities of XML data, Data and Knowledge Engineering, Vol. 60, No. 2, pp. 361-376.

[12] Tran, T. Nayak, R. and Bruza, P., 2008. Combining structure and content similarities for XML document clustering, In Proceedings of the 7th Australasian Data Mining Conference, Vol. 87, Australian Computer Society, Inc, pp. 219-225.

[13] Kim, W., 2008. XML document similarity measure in terms of the structure and contents, In Proceedings of the International Conference on Computer Engineering and Applications (CEA 2008), pp. 205-21.

[14] Viyanon, W. Madria, S. K. and Bhowmick, S. S., 2008. XML data integration based on content and structure similarity using keys, In On the Move to Meaningful Internet Systems: OTM, Springer Berlin Heidelberg, pp. 484-493.

[15] Dalamagas, T. Cheng,T. Winkel,K. J. and Sellis, T., 2006. A methodology for clustering XML documents by structure, Information Systems, Vol. 31, No. 3, pp. 187-228.

[16] Lian, W. Cheung, D. L. Mamoulis, N. and Yiu, S. M., 2004. An efficient and scalable algorithm for clustering XML documents by structure, Knowledge and Data Engineering, IEEE Transactions on, Vol. 16, No. 1, pp. 82-96.

[17] Zhao, Y. and Karypis, G., 2001. Criterion functions for document clustering: Experiments and analysis, Technical report, pp. 01-40.

BIOGRAPHY

Somayeh Ghazanfari received master science of computer engineering in 2015 from Islamic Azad University of Lorestan, Iran. Her current research area is data mining and specially clustering.

Hassan Naderi received his PhD degree in 2006 from INSA-LYON university of France. His current research areas are text mining, search engine and massive data processing.

This paper may be cited as:

Ghazanfari, S. and Naderi, H., 2015. "A Hybrid Algorithm for Improvement of XML Documents Clustering". *International Journal of Computer Science and Business Informatics, Vol. 15, No. 3, pp. 1-15*.