Clustering XML Documents using Structural Similarity

Moet Moet Lwin, Ei Chaw Htoon
motmotlwn@gmail.com, htoon.eichaw@gmail.com
University of Computer Studies, Yangon

Abstract

Extensible Mark-up Language (XML) is increasingly important in data exchange and information management. The automatic processing and management of XML-based data are ever more popular research issues due to the increasing abundant use of XML, especially on the web. Clustering is also helpful for categorizing web documents. Clustering, which means the physical arrangement of objects, can be an important factor in improving the performance in the storage model.

Clustering XML documents using structural similarity based on Progressively Clustering XML by Structural Similarity (PCXSS) method is presented in this paper. The PCXSS method intends to deal with the heterogeneous XML schemas to cluster XML documents by considering only the structural similarity. The efficiency of PCXSS methodology has been analysed with the real datasets which are ACM SIGMOD record, DBLP, XML Repository and Wisconsin's XML data bank.

Keywords: Clustering, PCXSS Methodology

1. Introduction

With the continuous growth of XML data, many issues concerning with the management of large XML data sources have also arisen. XML processing is a problem of continuing interest. XML is also becoming a common way of storing data. For efficient data management and retrieval, a possible solution is to group XML documents based on their structure and content. The element tags and their position in the document's hierarchy provide valuable information to clustering XML documents.
The clustering of XML documents facilitates a number of applications such as improved information retrieval, document classification analysis, structure summary, improved query processing and so on.The clustering process categorizes the XML data based on a similarity measure. Clustering of XML documents is more challenging because an XML document has a hierarchical structure and there exist relationships between element objects at various levels. [5]

The proposed system is composed of two main phases: tree merging phase and tree clustering phase. The rest of the paper is organized as follows: Session 2 will discuss the theoretical background, Session 3 the related work of the paper will be discussed, the design and implementation are presented in Session 4, Session 5 describes the experimental analysis of PCXSS clustering methodology and Session 6 concludes the paper.

2. Theoretical Background

The PCXSS method is an incremental clustering method that computes the similarity between a new XML document and existing clusters by considering the structures within documents. Figure 1 illustrates a high level view of the PCXSS method. The *pre-processing phase* decomposes every XML document into the structured path information called node paths. Each path contains the node properties from the root node to the leaf node. The *clustering phase* consists of two stages: structure matching and clustering. At structure matching stage, the similarity between an XML document and existing clusters is measured. The output of this stage is a similarity value called Common Path Similarity (CPSim) between an XML document and a cluster. CPSim is then used in the clustering stage to group the XML document into an existing cluster if CPSim exceeds the clustering threshold and the cluster has the largest CPSim with the tree else it is assigned to a new cluster.

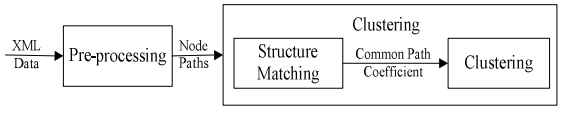

Figure 1 - The PCXSS Methodology

The structure matching stage uses two types of matching: tree to tree matching between two trees and tree to cluster matching between a tree and a cluster. The tree to tree matching is the matching between a new tree and a cluster that contains only one tree. This is defined as:

$$CPSim(Tree_1, Tree_2) = \frac{\sum_{i,j=1}^{|TPath_1|} \max(Psim(P_i, P_j))}{Max(|TPath_1|, |TPath_2|)} \quad \ldots. \text{ Eq (1)}$$

CPSim is the common path similarity between two XML trees. The CPSim of trees, $Tree_1$ and $Tree_2$ is the sum of the best path similar coefficient (PSim) of paths, P_i and P_j with respect to the maximum number of paths, $|TPath_1|$ and $|TPath_2|$ of trees, $Tree_1$ and $Tree_2$, respectively.

The tree to cluster matching is the matching between a new tree and the common paths in a cluster. It is defined as:

$$CPSim(Tree, Cluster) = \frac{\sum_{i,j=1}^{|TPath|} \max(Psim(P_i, P_j))}{Max(|TPath|)} \quad \text{.... Eq (2)}$$

CPSim between a tree and a cluster is the sum of the best PSim of paths, P_i and P_j with respect to the number of paths, |TPath| in the Tree.

An incremental clustering method is used in clustering stage of PCXSS. It first starts off with no cluster. When a new tree comes in, it is assigned to a new cluster. When a next tree comes in, CPSim is computed between the tree and the existing cluster.
If CPSim exceeds the clustering threshold and the cluster has the largest CPSim with the tree then the tree is assigned to that cluster else it is assigned to a new cluster. [5]

3. Related Work

Clustering XML data is more complicated than common text data as XML allows inserting structural and conceptual aspects into document content. Clustering of XML documents involves consideration of two document input features, namely structure and content, for determining the similarity between them. Most of the existing approaches do not focus on utilizing these two features due to increased computational storage and processing.

Ho-pong Leung et al. [1] proposed a novel XML structural representation called CXP. CXP encodes the frequently occurring elements with the hierarchical information to form the feature vectors for XML document clustering. And then, the clustering method which used path similarities between data nodes was presented by Ilhwan Choia et al. [2]. Good performance in query processing is provided by using path similarities because it can reduce page I/Os required for query processing. Jeong Hee Hwang and Keun Ho Ryu [3] discussed that the approach first extracts the frequent structures from XML documents and then, a new XML document clustering algorithm using common structures, which does not use measure of pair wise similarity between XML documents is performed.

Also, an experimental study which conducted over the INEX 2008 Document Mining Challenge corpus using both the structure and the content of XML documents for clustering them was described by Sangeetha Kutty et al. [4]. The concise common substructures known as the closed frequent sub trees are generated using the structural information of the XML documents. Again, a similarity measure (CPSim) between an XML document and a cluster of XML documents was defined by Tien Tran and Richi Nayak [5]. This measure was then used by an incremental clustering algorithm called PCXSS. The results and experiments were performed on the INEX 2006 Document Mining Challenge Corpus with the PCXSS clustering method.

4. Design and Implementation

The overall system design is illustrated in Figure 2. The input XML documents are parsed with Document Object Model (DOM) parser to validate whether they are well-formed or not. The DOM loads and parses the XML data into a tree structure. After the validation of the XML documents, merging phase will be performed.

Merging Phase decomposes XML tress into the structured path information called node paths. And then duplicated node paths in each document structure are eliminated. After the merging phase of XML documents, documents are represented as a collection of distinct node paths. And then, these common tree paths are used as input for clustering them using PCXSS method.

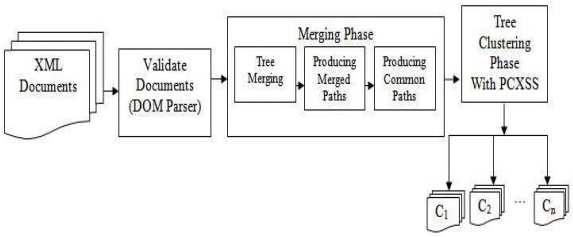

Figure 2 - System Design

Clustering Phase consists of two stages: structure matching stage and clustering stage. At the structure matching stage, the similarity between an XML document and existing clusters is measured. The output of this stage is a similarity value called CPSim between two trees and between an XML document and a cluster. CPSim is then used in the clustering stage to group the XML document into an existing cluster if CPSim exceeds the clustering threshold and the cluster has the largest CPSim with the tree else it is assigned to a new cluster.

The flowchart of the system is depicted in Figure 3. This system accepts the input XML files from user. After validation and decomposition of path structures from the XML file, it is saved into disk file. These steps continue until the user stops input processing. Then these structured tree paths are merged into one and saved into the database in order to retrieve the common path structures in future. These common paths and user input XML trees are built into tree by common path matrix to calculate the CPSim values by using path similarity calculation algorithm. With

respect to the proper threshold, an undefined number of clusters will be resulted by using PCXSS clustering method.

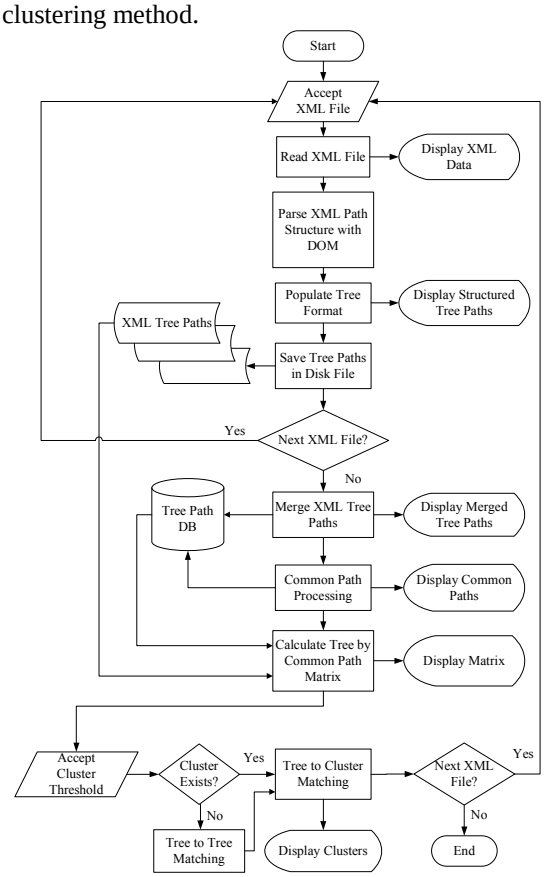

Figure 3 - System Flowchart

The structural similarity between two XML documents is measured by finding the common paths between two trees. In *common paths finding*, the degree of similarity between two paths, defined as PSim, is measured by considering the common paths between two trees. Figure 4 is used to calculate PSim value in this system.

```
PSim:=0;
  For each i in Tree-1 paths
    While j not end of Tree-2 paths
    If Path (p_i, p_j) = = 1
       PSim + = 1;
       j ++;
         break from while loop;
    Else
         j++;
           break from while loop;
    End If
   End While
 End For
```

Figure 4 - Path Similarity Calculation Algorithm

Table 1, 2 and 3 illustrate as an example of the processing steps of the system.

Table 1 Input Tree Paths

Input Trees	Tree Paths
url7.xml	urlset/url/loc urlset/url/lastmod urlset/url/priority urlset/url/changefreq urlset/url/loc urlset/url/lastmod urlset/url/priority urlset/url/changefreq
mondial11.xml	mondial/mountain mondial/desert mondial/island mondial/river/located mondial/sea mondial/lake
REC.xml	dblp/www/title dblp/www/url

Table 2 Merged and Common Paths

Merged Paths	Common Paths
dblp/www/title dblp/www/url mondial/mountain mondial/desert mondial/island mondial/river/located mondial/sea mondial/lake urlset/url/loc urlset/url/lastmod urlset/url/priority urlset/url/changefreq urlset/url/loc urlset/url/lastmod urlset/url/priority urlset/url/changefreq	dblp/www/title dblp/www/url mondial/desert mondial/island mondial/lake mondial/mountain mondial/river/located mondial/sea urlset/url/changefreq urlset/url/lastmod urlset/url/loc urlset/url/priority

Table 3 Tree Matrix

XML Tree	Path 1	Path 2	Path 3	Path 4	Path 5	Path 6
Tree1						
Tree2			1	1	1	1
Tree3	1	1				

XML Tree	Path 7	Path 8	Path 9	Path 10	Path 11	Path 12
Tree1			1	1	1	1
Tree2	1	1				
Tree3						

5. Experimental Analysis of PCXSS Clustering Methodology

PCXSS clustering algorithm and the evaluation criteria are implemented with Microsoft Visual Studio 2008 platform using C# language. All the experiments are tested on Intel® Core™2 Duo CPU

computer with 2.00 GHz processor, 4.00 GB memory (RAM) and Windows Vista. All the experiments in this paper are performed on four datasets [6], [7], [8] and [9] which is described in Table 4. The analyzed results are tested with 0.1 to 0.9 threshold values. The resulted number of clusters totally depends on the number of common paths and threshold value. Much the number of common paths, it becomes lesser the threshold value and also vice visa. If the threshold value is larger than 0.3, it is very sensitive to cluster input XML documents. When CPSim value is compared with this threshold value in clustering stage, input XML document is assigned into new cluster sensitively without necessary. Therefore, the threshold value 0.3 (optimal threshold value) is the most suitable in this system to get accurate cluster results. Figure 5 depicts the comparison of the execution time on different XML datasets. As a result, this system benefits the clustering time and accurate clusters of the input XML documents with threshold value 0.3.

Table 4 Datasets for Clustering

Datasets	No. of Documents	No. of Clusters
ACM SIGMOD Record	999	5
DBLP	290	6
XML Repository	120	25
Wisconsin's XML data bank(Niagara)	265	11
Heterogeneous Data	209	42

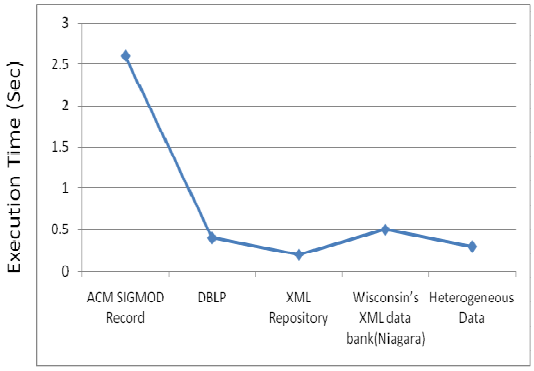

Figure 5 - Analysis Results for Clustering

6. Conclusion

The proposed system applies the PCXSS clustering method which considers only the structure similarity to cluster the XML documents. XML documents clustering are useful to XML application such as XML search engine. There are two advantages in this system: its linear time and its quality of clustering. While this system uses XML documents in ever-increasing web as input, PCXSS clustering method is suitable. Finally, as the role of XML documents becomes more important, PCXSS algorithm would be useful for clustering XML documents especially in search engines.

References

[1] Ho-pong Leung, Fu-lai Chung, Stephen C.F. Chan, Robert Luk, "XML Document Clustering Using Common XPath", Web Information Retrieval and Integration IEEE, Hong Kong, China, 2005.

[2] Ilhwan Choia, Bongki Moon b, Hyoung-Joo Kim, "A clustering method based on path similarities of XML data", Korea, 2006.

[3] Jeong Hee Hwang and Keun Ho Ryu, "Clustering and Retrieval of XML Documents by Structure", Korea, 2005, pp. 925 – 935.

[4] Sangeetha Kutty, Tien Tran, Richi Nayak, and Yuefeng Li, "Clustering XML documents using frequent subtrees", Brisbane, Australia, 2009.

[5] Tien Tran and Richi Nayak, "Evaluating the Performance of XML Document Clustering by Structure Only", In: Proceedings of the Initiative for the Evaluation of XML Retrieval, Springer Verlag, Brisbane, Australia, 2007, pp. 473-484.

[6] http://www.informatik.unitrier.de/~ley/db/about/dblp

[7] http://www.acm.org/sigmod/record/xml.

[8] The XML data repository. Accessed from: http://www.cs.washington.edu/research/xmldatasets/, Cited Sept 2004.

[9] http://www.cs.wisc.edu/niagara/data.html/. Niagara query engine.