# Redundancy recognition in heavy weight structure with different parameters

S. Sahunthala*

Department of Information Technology,
Anand Institute of Higher Technology,
Kazipattur, Chennai – 603103, India
Email: sahukalai2011@gmail.com
*Corresponding author

A. Udhaya Kumar

Department of MCA,
Hindustan Institute of Technology and Science,
Padur, Chennai – 603103, India
Email: aukumar71@gmail.com

Latha Parthiban

Department Computer Science,
Pondicherry University,
Kalapet, Puducherry – 605014, India
Email: lathaparthiban@yahoo.com

**Abstract:** Volume of data are transferred on the internet with the user defined format. Replica data and heavy weight structure process turn into problem if data is being processed in data ware house. It degrades the performance of query processing and occupies extra memory space. This paper analysis the data replica detection in the heavy weight structure with different parameters. In the existing techniques heavy weight structure is used to find the replica detection. If the number computation is increased, the performance is decreased when the query is processed. If the heavy weight structure absorbs huge amount of space it takes more time for query processing. The proposed technique – light weight binary duplicate detection (LWBDD) helps to get better outcome for query processing with detection of replica data in a hierarchical structure. This technique also support to generate good quality outcome than the existing approaches.

**Keywords:** replica detection; properties; heavy weight structure; light weight structure; query process.

**Reference** to this paper should be made as follows: Sahunthala, S., Udhaya Kumar, A. and Parthiban, L. (2023) 'Redundancy recognition in heavy weight structure with different parameters', *Int. J. Advanced Intelligence Paradigms*, Vol. 24, Nos. 1/2, pp.145–155.

**Biographical notes:** S. Sahunthala is currently an Associate Professor in the Department of Information Technology at Anand Institute of Higher Technology, India.

A. Udhaya Kumar is currently a Professor in the Department of Master of Computer Applications at the Hindustan Institute of Technology and Science India. He had completed his PhD in the area of Stochastic Optimization.

Latha Parthiban is currently a Professor in the Department of Computer Science at Pondicherry University India. She had completed her PhD in the area of Data Mining. Her research areas are data mining and image processing.

## 1 Introduction

Nowadays the World Wide Web transfer the data through the internet. In early days heavy weight structure data plays a vital role in the World Wide Web. A volume of data is travelling through the internet. The data is delivered in the form of heavy weight structure and it supports its own markup language for the development of the application and provides portability to carry the data. Heavy weight structure XML has the technique to transfer and publish the data on the web with respect to the business, application, etc. In real world, data cleanup and repeated data detection in the application is the vital task to transfer the data among the business. Heavy weight structure struggles to handle all type of data in hierarchical structure. The XML structure is symbolised by the format of the tree structure. The redundancy detection identifies more identical entities (Weis and Naumann, 2004) in the data. The main application of duplicate detection and removal is data cleanup and data incorporation. This paper supports the user who is involved in the data processing of hierarchical data and heavy weight structure data. JSON or JavaScript Object Notation is a lightweight text-based open standard. This format is designed for human-readable data interchange. JSON contains no tags to represent the hierarchical data. Elements are defined serially in the hierarchical structure. Data can be interpreted easily if we are using light weight format structure. It makes easier to parse the element from the hierarchical structure. It is syntax for storing and interchanging the data.

In real world, heavy volume of data will be stored from the data warehouse. If it occupies we can able to develop the process of query efficiently from the heavy weight structure. The role of replica data, how it is recognised and how the performance is happened in heavy weight structure is given in Naumann and Herschel (2010).

In this paper, Section 2 illustrate the literature survey of the different methods present to discover the replica data in heavy weight structure, Section 3 illustrate the issues recognised in existing approach and proposed the technique to improve the recognition of replica data and decrease the memory space in heavy weight structure and Section 4 shows the conclusion of the paper.

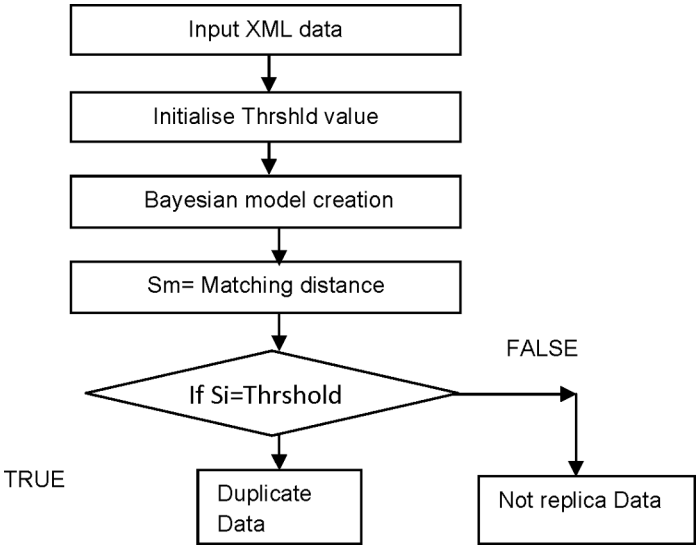
## 2 Literature review

We focus on the issues of recognition of replica data in hierarchical heavy weight structure with different parameters such as properties, entities etc. This part illustrates different methods to identify the replica data in heavy weight structure. The structured format data such as tuple replica recognition is analysed in Elmagarmid et al. (2007) and Naumann and Herschel (2010).

### *2.1 XML dup system*

This XML dup system (Leitão et al., 2013) is designed based on the Bayesian network model. The distance between two entities are calculated by using the overlay technique. If $W$ and $X$ are two heavy weight structure then the overlay between $W$ and $X$ are computed if and only if the element have the same path from the root element. In this technique, the primary threshold value is assigned based on the heavy weight structure. The workflow of the XMLdup system is shown in Figure 1.

**Figure 1** Workflow of XMLdup technique




The following probabilities are use to position the replica elements in two heavy weight hierarchical structure with the Bayesian network.

*Prior probability*: The default constant $fi_a = \text{simi}(V_l[y], V_m[y])$ is assigned depends on the similarity value between two elements, where $y$ is the attribute value of $l$th and $m$th attribute in the heavy weight structure.

*Conditional probability*: It defines the probabilities between two elements by the conditional probability $P = (Ei \mid E1, E2, \ldots, En)$ of the path from the root element to the sibling element node in the heavy weight structure.

*Final probability*: This probability will find the replica data in heavy weight hierarchical structure. The final probability $Fi$ is given as

$$Fi = X - \frac{P(M1) + P(M2) + \cdots + P(Mn)}{n}$$

where the element node score value is $M$, $w$ is the non-leaf element node value from the root element node, $n$ is the number of attribute value in the heavy weight structure.

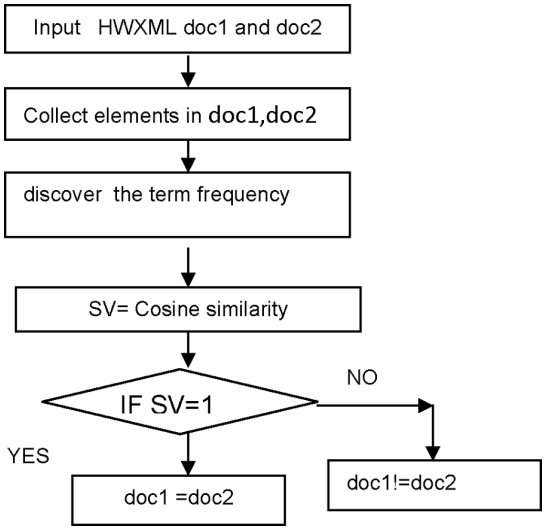
*Limitations*: Without machine learning and light weight structure approach the query is processed for a long time in the heavy weight hierarchical structure.

## 2.2 Xml join technique

In heavy weight XML structure a join operation is suggested to find the replica data. Tree edit distance (Leitão et al., 2013) is used in heavy weight structure join algorithm. Cosine method is used to compare two entities.

The flow of Xml join operation is shown in Figure 2. The comparison is done based on heavy weight XML structure. The similarities between two elements are calculated by Cosine similarity approach. It is denoted as $\text{Simdoc}(\text{doc1}, \text{doc2}) = \sum A_k \sum B_l \Big/ \sum \sqrt{A_k^2} \sqrt{B_l^2}$ where ($k = 1$ to $m$, $l = 1$ to $m$) doc1 and doc2 are two heavy weight hierarchical documents, $A$ and $B$ are hierarchical tree structure, $A_k$ and $B_l$ are the value of the elements $A$ and $B$, respectively.

**Figure 2** Flow of Xml join technique




*Limitations*: This technique is used to compare the simple heavy weight structure. When we try to process the complex heavy weight structures the query is processed with slow manner. It arises the less performance of the query processing in the heavy structure.

## 2.3 Dogmatrix framework

Dogmatrix technology (Weis and Naumann, 2005) uses keys to evaluate the entities in the heavy weight structure. It evaluate the elements not only based on

the direct values also based on their parents, children and heavy weight structures. Hence the replica objects are formed together by using recognition of the unique key. The unique key is generated for each object in heavy weight structure. ToXGene tool is used to assign a single Identification for each objects. Replica detection framework contains three parts. First part is candidate definition. Replica definition is the second part. Replica detection is the third part. Candidate definition part constructs the related objects to be compared in the heavy weight structure. Replica definition recognises replica objects based on their depiction. Replica detection part identifies the approach to find the entity detection. This technique track the Query formulation and effecting of replica candidates from all possible heavy weight structure candidates based on their information.

*Limitations*: When the heterogeneous heavy weight structure is used it arises less performance in processing the query in the replica detection structure.

### *2.4 SoXMLNeighbour technique*

This approach (Puhlmann et al., 2006) analyse the similarity between entities in a heavy weight structure. The entity depiction of the entity is symbolised by OBJDESC1, OBJDESC2, …, OBJDESC*n*. Hence the similarity between entities is attained by with different objects. The entity evaluation is done by OBJDTDIS as follows

$$(odt_k, odt_l) = \{1 \text{ if } m_k \text{ and } m_l \text{ not computable in comparison}$$

The above distance is derived from ned($w_i$, $w_j$) computation. In this method key is the main role identify the replica data in heavy weight hierarchical structure. This method has two stages. One is key production for each entity another one is key pattern detection part. If the keys are arranged accurately then it generates good outcome. If the keys are not arranged accurately and contains error then it generates worst outcome in the heavy weight structure replica recognition of data.

*Limitations*: Not guaranteed process that all objects are compared or not in the heavy weight structure.

### *2.5 Domain independent approach*

Transitive closure computation is used to group the elements in this approach (Kalashnikov and Mehrotra, 2006). String similarity computation is used to compare the elements. The number of computation is reduced by filter procedure.

This approach is used to locate the replica elements in the simple heavy weight structure. Let UU and UU[1] are two nodes from doct 1 and doct 2, respectively. The elements are repeated by one of the following definitions

- if the parent elements of UU and UU[1] are same
- the identical name of UU and UU[1]
- children nodes of UU and UU[1] are equal structure or contains the equal data
- the string comparison calculation is based on the Inverse document frequency (IDFt)

$$\log \frac{M}{tf}$$

where $M$ is the number of heavy weight structure. $tf$ is the number of rate of the element in all heavy weight structures.. Two string $St1$ and $St2$ similarity computation is done by

$$simi(St1, St2) = \frac{IDFt(St1 \cap St2)}{IDFt(St1 \cup St2) / IDFt(St1 \cap St2)}$$

It has filtering method to reduces the computational complexity in heavy weight structure. Some filtering methods are:

*Length dis filter*

Let str1 and str2 are two strings. If length of two strings are length(str1), length(str2) it is true when the following computation holds.

$$length(str1) - length(str2) | \leq dist(str1, str2)$$

*Bag dis filter*

It is used to evaluate three strings. Let $X$, $Y$, $Z$ are be the three strings from the heave weight structure the comparison is computed by

$$dist(X, Y) + dist(Y, Z) \geq dist(X, Y) \geq | dist(X, Y)$$

This is based on the meaning of the string. The another approach to detect the distance among three strings $X$, $Y$, $Z$ are

$$distbg(X, Y) = \max(| X, Y |, | Y - Z |)$$

*Limitations*: Difficulty arises in the process of integration of more than one heavy weight structure and consumption of memory space is high.

### *2.6 Xedge index algorithm*

In Xedge Index technique the heavy weight hierarchical structure similarity is measured by the index level of the structure to be compared. In each level the element is recognised with the unique number. In this method the unique number and different terms find in each level of the heavy weight hierarchical structure. The string edit distance method is used to perform whether the terms are similar or not. This technique is detailed in Antonellis et al. (2008).

*Limitations*: If level of the heavy weight structure is increased then performance is less in processing the query via the level of the hierarchical structure.

## 3 Problems in existing methods and proposed approach

In existing replica detection approaches we analysed all elements in the heavy weight structure with different parameters. We propose the method of LWBDD to discover the replica data in heavy weight structure. The proposed method is used with light weight process as JSON to find the repeated elements in the structure. Hence we discussed the

problems in existing techniques and also how we can increase more performance in proposed approach.

In XMLdup system if the system is complex we can't able to detect all possible repeated data. We cannot receive high precision and recall in the query processing in heavy weight structure when the threshold user defined value is very low. We can design the Bayesian network with machine learning approach and Light weight format. The process of repeated element detection is faster. Cosine method arises poor recital to detect match between two entities. The comparison time is increased when the heavy weight structure is very huge. Hence the XML structure is the heavy weight process. *N* gram value is projected to enhance the performance of the query processing. Each *N* gram is allotted to specific window. This window takes more space in heavy weight structure. enhancing the parameter of *N* gram value. The hierarchical document structure can be light weight format of JSON. Dog Matrix issues are defined in Section 2.7. These issues are overcome by using the light weight structure with dog matrix framework.

Two problems are arrived by using the technique SXNX approach to detect replica elements in heavy weight structure. First problem is find the same elements in all group of elements second problem is whether every elements are compared with all possible group of elements. If the automatic key is produced with the light weight structure (JSON) we can able to get good result for comparison of objects in ToXGene tool. In the two level optimisation (Leitão and Calado, 2013) technique the following problems are occurred. It takes more computation for comparison between elements in heavy weight hierarchical structure. So the cost is high and occupies more space in computation of comparison of objects and degrades the performance.

The heavy weight same structure is identified by two tier index (Sun et al., 2014) tree. It occupies more memory space to compare the objects due to the creation of the heavy weight document entry with the index structure node. The query processing time is automatically decreased. If we are using light weight structure with the above mentioned approach we can able to produce accurate processing time in the query analysing process of heavy weight structure. Xedge algorithm [11] uses the index level to compare the nodes on both the heavy weight structure. The structure with the homogeneous can be used in this method. Hence the similarity parameter cosine or Jaccord distance is not suitable for heterogeneous structure with the light weight format. It will be produced by using the index as the Bayesian level. Table 1 shows the issues in various existing techniques and the method to be proposed to find the duplicate data and reduces the memory space in hierarchical structure to overcome the issues.

The proposed technique to reduce the time to process the data in the hierarchical structure to analyse the duplicate data. Hence the method light weight binary duplicate detection (LWBDD) is used to find sample documents with the light weight format.

The proposed method work as follows the algorithm of LWBDD is shown in Figure 3.

We analyse the different similarity measure approach with the documents in Figures 4 and 5 with respect to the number of computation and light weight process. The computation may be the basic operation such as addition, multiplication, division etc and the light weight process of JSON format. If the number of computation is reduced, it increases the speed of the process in the data structure. If we are using light weight process then the space is reduced to process the query in the hierarchical structure. Document 1 and Document 2 are converted into JSON format. In Figure 4, the elements

are extracted as Person, title, Name(Firstname, Lastname) Awards and Year. In Figure 5, the elements are extracted as in Figure 4. Then document 1 (Figure 4) element in the each level is compared with the element which can be positioned on the document 2 (Figure 5). If the result is same the binary calculation is performed. At the end of the comparison if the result is not affected we conclude both documents are similar.

**Figure 3** Proposed system model

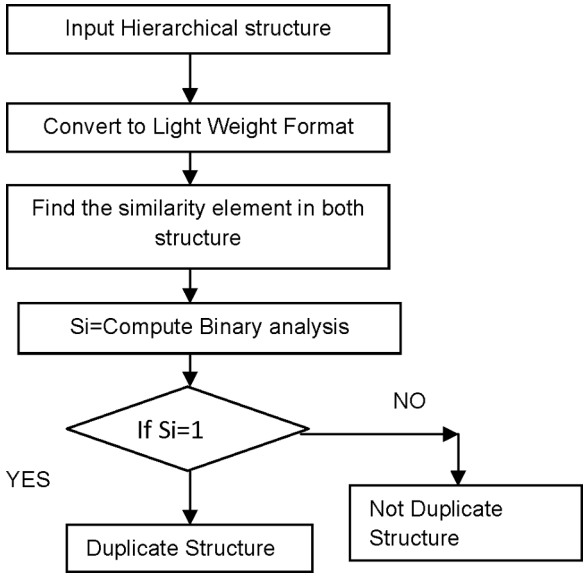



**Figure 4** Document 1

```
<Person>
<title></title>
<Name>
        <Firstname></Firstname>
        <Lastname></Lastname>
</Name>
<Awards></Awards>
<Year></Year>
</Person>
```

**Figure 5** Document 2

```
<Person>
<title></title>
<Name>
        <Firstname></Firstname>
        <Lastname></Lastname>
</Name>
<Awards></Awards>
<Year></Year>
</Person>
```

**Figure 6** Light weight binary similarity of hierarchical structure

Algorithm : Binary similarity of Light Weight hierarchical struct

Input : Hierarchical documents

Variables : m is the no of documents, n is the number of elements in mth document and p is the number of elements in the m+1th document

Output : Binary value

Steps :

1. Give input as Hierarchical documents
2. Create Light Weight format for all input
3. Extract the element in all light weight documents
4. Find the number of elements
5. Loop x=1 to s docs
6. Loop m=1 to n lines
7. Loop n= 1 to p elements
8. T=Perform similarity(m,n)
9. If T=0 , break
10. End loop,End loop,End loop
11. If (T=1)
12. Similar heavy weight structure
13. Else
14. Not similar heavy weight structure

**Table 1** Issues in the existing technique and method to be proposed to find the replica data in hierarchical structure

| *S. No.* | *Existing method* | *Performance issues* | *No of computation* | *Status of light weight process* |
|---|---|---|---|---|
| 1 | Xmldup | Query is processed by long time | M-1 | Not Used |
| 2 | Xml join technique | Used to compare the simple structure | 3M+1 | Not Used |
| 3 | Dogmatix | Less performance in the heterogeneous structure | $M-1$($M$ is the number of object description) | Not Used |
| 4 | SXNM | Not guaranteed to compare the entities | $M-1$ | Not Used |
| 5 | Two level optimisation method | Cost is high for computation in comparison of objects | $2M-1$ | Not Used |
| 6 | Domain Independent method | Problem in integration of more than one document and utilisation of memory is high | $3M-1$ | Not Used |
| 7 | Two tier index structure | Occupies more memory space when comparing the objects | $M-1$ | Not Used |
| 8 | Xedge index algorithm | If level of the structure is increased then less performance in processing the query | 4M + 3 | Not Used |
| 9 | LWBDD method | Element extraction may be complicated when huge amount of data is used | 1 | Used |

Table 1 *M* is the number of elements in heavy weight structure.

Table 2 shows the number of computation with our sample structure given in Figures 4 and 5. The method LWBDD is better than the existing techniques which can be discussed in Section 2.

**Table 2** No. of computation in different techniques with sample documents in Figures 4 and 5

| *S. No.* | *Technique* | *No. of computation* |
|---|---|---|
| 1 | Xmldup | 6 |
| 2 | Xml join technique | 22 |
| 3 | Dogmatix | 6 |
| 4 | SXNM | 6 |
| 5 | Two level optimisation method | 13 |
| 6 | Domain independent method | 22 |
| 7 | Two tier index structure | 6 |
| 8 | Xedge index algorithm | 31 |
| 9 | LWBDD method | 1 |

## 4 Conclusions

LWBDD works with various heavy weight hierarchical structure which helps in recognising replica detection methods with different parameters. The parameters are such as properties, entities, structure of heavy weight tree format. The hierarchical heavy weight structure is converted into light weight structure. This structure occupies less memory space. The study of different techniques in replica identification of between elements from heavy weight structure shows the occupation of memory is high. If that structure is changed into Less Weight Structure the outcome of the result is better than the existing technique.

## References

Antonellis, P., Makris, C. and Tsirakis, N. (2008) 'XEdge: clustering homogeneous and heterogeneous XML documents using edge summaries', *Proceedings of the 2008 ACM Symposium on Applied Computing*, March, ACM, pp.1081–1088.

Elmagarmid, A.K., Ipeirotis, P.G. and Verykios, V.S. (2007) 'Duplicate record detection: a survey', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 19, No. 1, January, pp.1–16.

Kalashnikov, D.V. and Mehrotra, S. (2006) 'Domain-independent data cleaning via analysis of entity-relationship graph', *ACM Trans. Database Systems*, Vol. 31, No. 2, pp.716–767.

Leitão, L. and Calado, P. (2013) 'Efficient XML duplicate detection using an adaptive two-level optimization', *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, March, ACM, pp.832–837.

Leitão, L., Calado, P. and Herschel, M. (2013) 'Efficient and effective duplicate detection in hierarchical data', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 25, No. 5, May, pp.1028–1041.

Naumann, F. and Herschel, M. (2010) 'An introduction to duplicate detection', *Synthesis Lectures on Data Management*, Vol. 2, No. 1, pp.1–87.

Puhlmann, S., Weis, M. and Naumann, F. (2006) 'XML duplicate detection using sorted neighborhoods', *International Conference on Extending Database Technology*, March, Springer, Berlin, Heidelberg, pp.773–791.

Sun, W., Qin, Y., Wu, J., Zheng, B., Zhang, Z., Yu, P., Liu, P. and Zhang, J. (2014) 'Air indexing for on-demand XML data broadcast', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 25, No. 6, June, pp.1371–1381.

Weis, M. and Naumann, F. (2004) 'Detecting duplicate objects in XML documents', *Proceedings of the 2004 International Workshop on Information Quality in Information Systems*, June, ACM, pp.10–19.

Weis, M. and Naumann, F. (2005) 'DogmatiX tracks down duplicates in XML', *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, June, ACM, pp.431–442.