A Review on Coreference Detection in Xml Schema

[1]Aayushi A. Shah, [2]Dhatri Pandya
[1] PG student, [2]Assistant Professor
[1]Computer Engineering Department,
[1]Sarvajanik College of Engineering
and Technology, Surat, India

Abstract : In the modern era of technology and innovation, Data is generated and transferred in a large amount. Data can be proper or improper. Proper data means the data which is original and non-duplicated. This type of data rarely exists. Improper data can be thought of as duplicated data or redundant data which are of no use for processing or transferring. The reason is such data leads to hazardous problem and can give unwanted results. Data needs to be preprocessed or cleansed before its usage. The preprocessing involves detecting the quality of data and removing redundancy if exits. Such redundancies can be removed using coreferent or duplicate detection techniques available in Service Oriented Programming (SOA). SOA mainly deals with web services, protocols and distributed environment. The most important language which is used for SOA is Extensible Markup Language (XML). Although, XML allows exchange and publication of data on Web, there are obvious chances of inconsistencies and errors in data. Hence, there is a need for XML data cleansing. In this paper, different coreferent detection techniques i n XML schema are studied and compared. The useful feature based parameters are also discussed.

IndexTerms **- Service Oriented Programming, XML Schema, Coreferent Detection, Data Duplication.**

I. INTRODUCTION

Detecting duplicates is a difficulty in many domains like data warehousing and customer relationship management. The duplicate detection problem occurs in record linkage, entity identification and object matching in data. Different representation of a same object can be referred as duplicates. Data Mining, Customer Relationship Management and Data Warehouses require pre-processing steps before analyzing data. These pre-processing steps include duplicate detection and data cleansing. Likewise, Data Integration involves heterogeneous and distributed data sources which are combined to represent every data object into a complete, accurate and exclusive way. Another situation where duplicates occur and require to be identified is data integration in which data from distributed and heterogeneous data sources are combined into an exclusive, complete, and accurate representation for every object. Quality of data is lowered due to the presence of duplicate or coreferent data that defines the same entity but in a dissimilar way across related and multiple databases [1]. This type of data should be avoided for proper analysis of data [2,9]. This brings the need of preserving data quality [10]. Data duplication can happen because of errors or different modeling and representation. It is possible to detect and handle duplicate data through data or combination of data and metadata. XML is popularly used for exchange and publication of data on the Web [3,8]. The main attention in duplicate detection is analyzing duplication in the hierarchical and semi-structured type of XML data. These types of data strongly differ from structured relational model. XML data cleansing is necessary to ensure good quality of data in many scenarios. Duplicate Detection resolves the duplicate entries in the data which determines the same meaning in the real world. The other type of data is called Fuzzy data. Duplication in fuzzy data can be detected using more sophisticated techniques to find various inconsistencies, missing data or several spelling errors [3]. XMLDup is one of the methods to detect Duplicates. It uses a Bayesian network to determine the probability of two duplicate XML elements. Information within elements and its structure is considered in this technique [5]. We study Pruning method in this paper which is similar to XMLDup. The rest of the study is organized as follows: The second section represents existing methods; third section represents analysis of methods with their feature based parametric comparison in tabular format and fourth section gives conclusion of the paper.

II. EXISTING METHODS

There are many approaches to find duplicates in XML schema. We focus on different techniques to detect duplication efficiently and effectively. We describe five approaches hereby and the technical comparison is included in the next section.

A) Content Based Matching: This technique consists of lexical and statistical comparison of content data and coreferent tuples detected across different datasets. This possibly leads to match the schema in a correct way. It is assured that knowledge of slightly small number of duplicate tuples is enough for correct matching between corresponding attributes of different schemas [12]. This technique is a novel automatic and semantical matching of duplicate attributes in

schemas according to data and meta-data. It combines vertical and horizontal matching which thereafter apply Probabilistic Truth Values (PTVs) and cardinality of a set of PTVs and determines the uncertainty about the matchings. Figure 1 describes various steps of Horizontal and Vertical Schema matching. Vertical schema matching is the first phase in which statistical analysis of attribute domains of the schema is done. In the second step, attributes which does not possess similar statistical properties are considered and overlapped using some threshold. The last step of Generalization performs one to many schema matchings. Similarly in Phase II of horizontal schema matching, two steps are involved where duplicate rows are detected and schema matching is done.

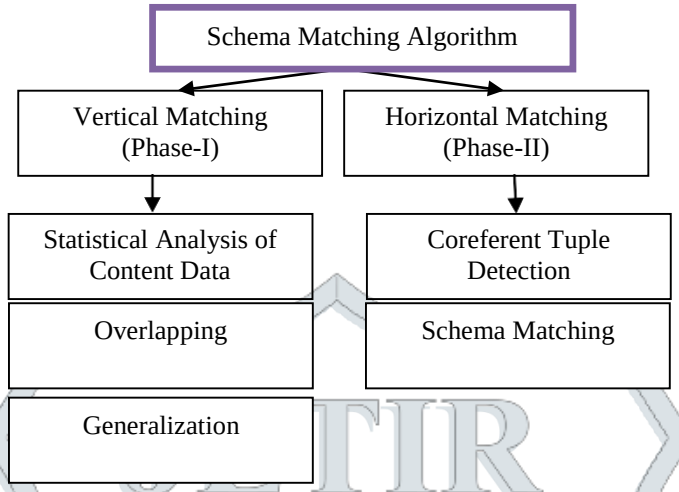

Fig.1 Content Based Schema Matching Algorithm

Apart from this, this approach helps to manage the problem of information coverage and attribute granularity. Comparison operator plays a major role in establishing the duplicate objects. It works at the level of both metadata and object features. Aggregation is another operator that combines the comparison scores that are obtained for particular features. Two objects are said to be coreferent when they determine the same real-world entity. Two thresholds namely true and false symbolized as T and F are taken to decide whether two objects are coreferent. If probability p(F) is lesser than the thresholdF , then coreference is said to have occurred. If p(T) is lower than the thresholdT, then no coreference exist. In other way if both the thresholds exceed then the duplication status is said to be unknown. This approach establishes semantical matchings of schema elements in the situations where the schema information-only-based methods does not work. Attribute names may even confuse the schema matching methods which are only based on schema information. Content data can be used as valuable source of information which can considerably improve schema matching. It is useful to find the correspondences between the schema elements based on schema information only which is difficult or impossible. This technique explicitly copes with the uncertainty of semantical one-to-many and one-to-one schema matchings. [1]

B) Path Based Matching*:* This metadata-based method is syntactical and automatic method for finding duplicate elements in XML schema [13]. The comparison of tags and paths are made for the finding coreferent elements in XML schemas. Important data are considered to be kept while detecting duplicates by involving some heuristics based on the intuition that the important elements are distinctive as well as required. The important elements should be closer to the root. Information such as cardinality, order of element and its depth is used. Matching of two schemas is possible even with small amount of information available. An XML schema contains metadata that determines the structure of an XML document that enables to reconstruct paths. After that, sequences of element names are created. These are linked to a root element along with leaf elements in XML documents.

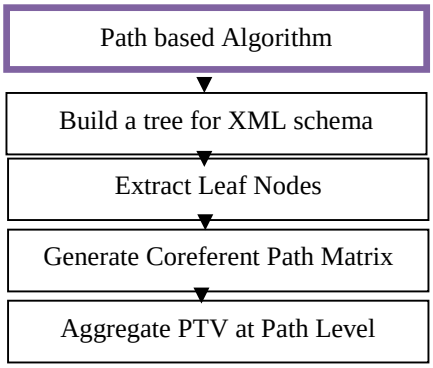

Fig. 2 Steps for Path Based Schema Matching

If the elements have different tags and are available at different levels of an XML file structure, it is necessary to identify duplication in them. First of all, input XML schema is structured as a tree. Later, paths are extracted for each leaf from a tree and this method generates matrix of coreferent path. Matrix represents one to one matchings between paths. Elements of the matrix states the degree of uncertainty about paths duplicates represented by PTVs. Various steps undertaken are Tokenization of a path, Step comparison, Mapping on steps level and Aggregation on steps level. At initial level, the technique determines coreference of steps. Then, the steps duplication information, with a clear representation of its related uncertainty using PTVs is aggregated properly to obtain data on paths duplicaton which is further aggregated to decide whether the XML schema contains coreference entities or not. This enables to explicitly manage the position of the elements and the relative importance of paths within their schema. The above steps concern about detecting similar elements in XML schemas. However, the last step is used only for detecting the similarity of overall XML schemas. Coreference path detection may further enable to determine coreference in XML schema. Paths are considered as the most important metadata in XML files. Detection of coreferent paths needs the knowledge of coreferent steps from which paths are constructed coreference is considered here in a hierarchical way. [2, 9]

C) **Fuzzy Based Detection:** This technique inspects on how object descriptions can be identified automatically. Automatic Duplication detection is mainly done for large data sets. This is a difficult task in XML as both the objects and object descriptions are represented by XML elements. Recent duplicate detection techniques consider not only data but also its structural diversity. Then comparison strategies are defined that make use of dependencies of elements and improves efficiency without compromising effectiveness. At last, scalability is considered by considering the fact that how different databases of relational type and XML types support the process of duplicate detection. This technique considers the problem of duplication for a wide range of applications. Description Selection is done to ensure about information that determines an XML element where XML element is considered as an object. A similarity classifier takes input as a pair of objects to classify them as non-duplicates or duplicates depending upon Object Dependencies (ODs). This is based on Domain-dependent classifiers that define thresholds for similarity values.

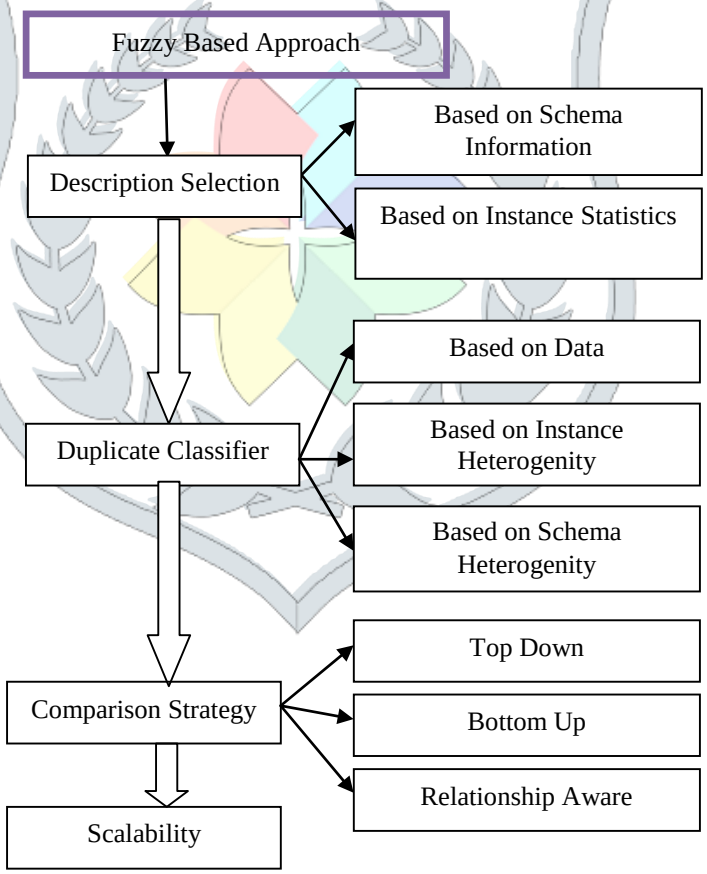

Fig. 3. Tasks for Fuzzy Based Schema Matching

Two heuristics for duplication detection considers the observation about the attributes. These attributes describe an object which is frequently defined in closeness of the XML element used to represent the object. It is known as object element. Statistics are used on the actual XML data along with schema information. Statistics are collected on the structure of the instance of XML document and may be on the content of XML elements. Relationships aware strategy is used for flexibility of top down and bottom up comparison of schema [3].

D) Improved Pruning: This approach uses the Bayesian Network to calculate the resemblance between two different XML objects that represent identical elements [7]. There exist two types of probabilities in the method namely Prior Probability and four different types of Conditional Probabilities. Prior Probabilities are defined as the probabilities of values being duplicates with respect to their parent XML node.

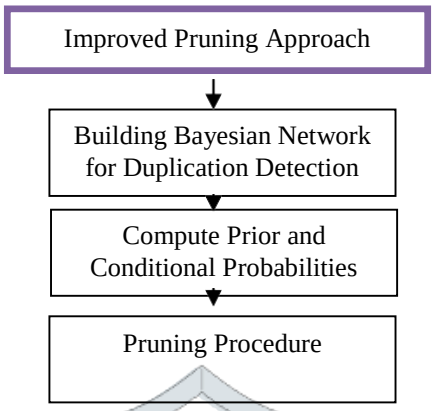

Fig. 4 Steps for Improved Pruning Approach

In conditional Probabilities, the probability of the values of the nodes being duplicates is based on assumption that each individual pair of values contains duplicate. The procedure begins with taking an input two nodes (N,T). N defines nodes and T defines predefined threshold value with which it is determined that given node is duplicate or not. The probability score for list of all parent node is taken 1. Then the next step includes computation of the definite probability value of each node of the parents of N. If a node n is a value node then the probability score is computed by finding the similarity of the values it represents. The selection attributes and nodes are chosen depending on the user at run time. On the other hand, if a node n is a not value node means it is child node then it is computed recursively until the leaf node is not found with updated threshold value with respect to that particular node. Once the score for node n is calculated, the total score for N is compared with threshold value and then it is decided to continue with the algorithm or stop the algorithm. Moreover, along with strategy to improve the efficiency and effectiveness, another special type of parameter is checked i.e typographical errors. To reduce or remove this error, two XML elements are compared and their blank or white spaces are removed [4].

E) Sorted Neighbourhood: It is used to cover not only a single relation but nested XML elements. XML parent and child relationships are used to compare objects.

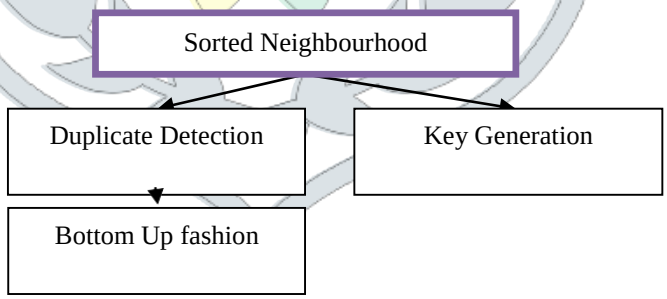

Fig. 5 Steps for Sorted Neighbourhood Approach

To ensure efficiency, windowing technique is applied in a bottom-up manner. Duplicates are detected at each level of the XML hierarchy. It is a very effective approach to detect coreference in a single relation and nested complex XML data. The first phase is known as key generation where key is generated. They are sorted lexicographically. In the second step, the elements are sorted using the keys generated and a sliding window is applied over the sorted elements. This step is called duplication detection phase. The elements are compared within the window. A window of fixed size slides over the sorted keys and searches coreferences only in the tuples referenced in the window. This limits the number of comparisons. The size of the window is important for the effectiveness of the algorithm and the quality of the result. Having a small window, only a small set of elements are compared. This leads to a relatively fast duplicate detection, though it has possibly poor recall. Having a large window results in a slower algorithm, however, the chance to find duplicates is better as more comparisons are performed. Relationships between different types of objects are broken by similarity measure, which considers duplicates among descendants. The Sorted Neighborhood Algorithm (SNA) is a well-known algorithm for the efficient detection of duplicates in relational data. If this approach is combined with new

approaches in duplicate detection, it becomes a realistic alternative for XML duplicate detection for huge amounts of data [5].

Table 1 Parametric analysis of XML Coreference Detection Approaches

Method / Parameter	Content data based [1]	Path based [2]	Fuzzy based [3]	Improved Network Pruning [4]	Sorted Neighborhoods [5]
Method used	Automatic detection using content data	Coreferent path	Schema statistics and information	Bayesian network and probability	Sliding window
Is efficiency considered?	Yes	Sometimes	Yes	Yes	Yes
Is scalability considered?	No	No	Yes	No	Yes
Datasets used	3	3	3	4	3
Based on	Metadata and data	Metadata	Data and structural diversity of xml	Data	Data
Strategy	Horizontal and vertical	Top down	Top down and Bottom up (relationship aware)	Top down	Bottom up
Comparison method	Statistical and lexical data comparison and threshold	Tokenization and step comparison	Similarity measure	Threshold	Threshold
Importance of elements considered?	No	Yes	Yes	No	Yes
Type of matching	Semantic	Syntactic	Synthetic	Probabilistic	-

III. Analysis of Existing Coreference Detection Approaches

There are different techniques to detect coreference in XML schema. In this section we analyze different parameters of coreferent detection techniques. Various parameters and their significance are discussed as under. The method used in each of the approaches play a very important role on how that approach works and what mechanisms they have used. To determine good coreference between the XML schema there is a need of two important parameters namely efficiency and scalability. The approach must be efficient and scalable enough to be used in different types of scenarios. The parsing strategy means the data is parsed or matched from top down or bottom up or both the direction. The method used for comparison is also reviewed which determines whether the data are compared statistically, tokenizing is required or not and some similarity measure or threshold is used for comparison. The importance of elements is considered in some approaches which is advantageous as duplication detection is done first for the important elements and then on the remaining elements or data. The type of matching also plays an important role and determines whether the approach is suitable for semantic, probabilistic or syntactic matching. We analyze the XML coreference detection approaches in this section in Table 1 where we compare the approaches with respect to different parameters that we discussed above in detail. This analysis is done on the basis of certain good and bad features of studied approaches. The above table showcases the parametric evaluation of five existing approaches. The table describes different parameters that can be used to evaluate discovery of resources. These approaches are considered in such a way that we can derive an idea how efficient they are to get optimized resource discovery.

IV. Issues and Challenges

The following are some of the challenges which are to be dealt for duplicating XML data.

- Data Overload: This is the main challenge as due to immense amount of data which are generated and transferred, it has become very challenging to find out duplicates.
- Structure of Data: Data can be proper data or improper data. It is nearly difficult to analyze different structure of data like structured, unstructured, hierarchical to name a few.

There are some issues which occur when XML Duplication is done.

- Complex Structure: XML has a very complex structure. Due to its complexity, it becomes difficult to detect duplication in it [11].
- Availability: This is an issue because in real XML data stored in various organizations are not available for public use due to privacy concerns. Due to unavailability of data and the organizations tend to share some portion of data which leads to unreliable analysis or duplication detection.

- Attribute Granularity: This can be defined as any entity whose data fields could be sub divided. This is also an important issue [1].

V. CONCLUSION

Data is generated in a large amount. Such data cannot be guaranteed correct or useful. Some cleansing needs to be done mainly for detecting and removing duplicated data. In case of Web services online, XML is widely used but still it contains some irrelevancy. Different approaches are reviewed and compared based on their feature based parameters. All the approaches have their own significance and own way of structuring XML schema and detecting duplicates. However, the coreference detection should be efficient, scalable and should be done even if little information is provided. This facility is provided by Fuzzy detection and Path based approach discussed earlier. If a hybrid approach of both is done to detect coreference then a more reliable detection can be done.

VI. REFERENCES

[1] M.Szymczak, A.Bronselaer, S.Zadro˙zny, Guy De Tré–Content Data Based Schema Matching|, Springer International Publishing Switzerland, 2016.
[2] M.Szymczak, S. Zadro_zny, A.Bronselaer,Guy De TrÃ, –Coreference detection in an XML schema|Information Sciences 296, pp237–262,Elseveir, 2015.
[3] M. Weis, –Fuzzy Duplicate Detection on XML Data|, Proceedings of the 31st VLDB Conference,Trondheim, Norway, 2005.
[4] V.BorateS.Giri, –XML Duplicate Detection with Improved Network Pruning Algorithm|, International Conference on Pervasive Computing (ICPC), 2015
[5] L.Leitão, P. Calado, M. Herschel,–Efficient and Effective Duplicate Detectionin Hierarchical Data|,IEEE Transactions On Knowledge And Data Engineering, Vol. 25, No. 5, 2013.
[6]S.Puhlmann, M. Weis, F.Naumann, –XML Duplicate DetectionUsing Sorted Neighborhoods|, Springer-Verlag Berlin Heidelberg, 2006.
[7] V. Kisan Borate, S.Giri, —Duplicate Detection in Hierarchical XML Data|, National Conference on Information Theory and Communication Networks (NCITCN-2014), pp. 218-221, 2014.
[8] Z. Bellahsene, A. Bonifati, E. Rahm (Eds.), Schema Matching and Mapping, Springer, 2011.
[9] M. Szymczak, S. Zadrozny, G. De Tré, Coreference detection in XML metadata, in: W. Pedrycz, M. Reformat (Eds.), Proceedings of 2013 Joint IFSA WorldCongress NAFIPS Annual Meeting, 2013.
[10] S. M., Bronselaer, A., Zadro˙zny, S., De Tré, G.: Semantical mappings of attributevalues for data integration. In: Proceedings of NAFIPS 2014. pp. 1–8. IEEE, 2014.
[11] A. Pradeep, T. George, –Duplicate Record Detection in XMLusing AI Techniques|International Journal of Computer Techniques, Volume 2 Issue 2, 2015.
[12] Zadro˙zny, S., Kacprzyk, J., Sobota, G,–Avoiding duplicate records in a database using a linguisticquantifier based aggregation—a practical approach|, Proceedings of FUZZ-IEEE, pp. 2194–2201, 2008.
[13] R. Nayak, T. Tran, –A progressive clustering algorithm to group the XML data by structural and semantic similarity|, IJPRAI 21 (4), 723–743, 2007.