Matching XML Documents in Hierarchical Data

[1]P.K.R. Madhuri, [2]Prof. Ch.Sita Kameswari
[1]Dept. of CSE from BABA Institute of Technology & Sciences, Visakhapatnam, AP, India
[2]Dept. of CSE, HOD & PG Coordinator, BABA Institute of Tech. & Sciences, Visakhapatnam, AP, India

Abstract
Duplicate detection consists in detecting multiple representations of a same real-world object, and that for every object represented in a data source. Duplicate detection is relevant in data cleaning and data integration applications and has been studied extensively for relational data describing a single type of object in a single table. This paper focuses on iterative duplicate detection in XML data. We consider detecting duplicates in multiple types of objects related to each other and devise methods adapted to semi-structured XML data. Relationships between different types of objects either form a hierarchical structure or a graph structure. Iterative duplicate detection require a similarity measure to compare pairs of object representations, called candidates, based on descriptive information of a candidate. The distinction between candidates and their description is not straightforward in XML, but we show that we can semi-automatically determine these descriptions using heuristics and conditions.

Keywords
XML Data, Hierarchical Data, Duplicate Detection

I. Introduction

The quality of the electronic data play a very important role in number of business processes, applications, and decisions. a study on data quality conducted in 2002 by the Data Warehousing Institute shows that data quality problems cost U.S. businesses more than 600 billion dollars a year. Poor data quality, which is the cause of all these problems, is due to several different types of errors. They are errors in a single data source and errors due to integration of several sources of data. Errors in data may be classified as errors on schema level and errors on data at data level. In this work , we deal with the problem of identifying duplicate representations of a same real-world object, an error occurring on data level both within a single source and in data integration scenarios. The challenge in duplicate detection is to detect duplicate representations that are not exactly equal due to errors in the data, and that cannot be identified using a universal identifier. Further errors are missing, outdated, or contradictory data. As ac consequence, duplicate detection cannot be performed just by checking on the equality of object attributes or global identifiers. Instead, more complex algorithms are required. Data cleaning consists in correcting errors and inconsistencies in data and is an issue of critical practical importance as it improves overall data quality. High data quality is the prerequisite for meaningful data analysis, required in scenarios such as report generation over data warehouses, customer relationship management, and data mining, to name just a few. When an object has multiple representations, analysis assumes that they are actually multiple different objects and therefore generates wrong results. Until recently, research on duplicate detection has focused on detecting duplicates within a single relational table. The schema of a table consists of several attributes. Every tuple in a relational table has exactly one value for every attribute. Most duplicate detection approaches designed for a single relation iteratively compare pairs of tuples as follows: They first compare attribute values pair wise by computing a value similarity, and then combine these similarities to a total tuple similarity. If the similarity is above a specified threshold, tuple pairs represent duplicates, otherwise they represent non-duplicates. We call this comparison approach a thresholded similarity measure approach, which is a popular approach for iterative duplicate detection in relational data. A good similarity measure is very important to find the correct duplicates. Another aspect that has been considered is the time complexity of an approach, i.e., efficiency, which is mainly improved by sophisticated tuple pruning techniques that avoid expensive similarity comparisons. To perform duplicate detection on large relational data not fitting in main memory, scalability has also been considered. Relational data only represents a fraction of today's data. XML is increasingly popular as data representation, especially for data published on the World Wide Web and data exchanged between organizations. In XML data, it is also true that different types of objects are described within a single schema, however, they are not necessarily described by a fixed set of single valued attributes due to the semi structured nature of XML. Consequently, similarity measures designed to compare equally structured flat tuples are no longer applicable to semi-structured hierarchical XML elements. Therefore, our second goal is to devise strategies for XML duplicate detection, which, in addition to considering relationships, also necessitate new techniques for similarity measurement. XML is used both for large scale electronic publishing of data, and for the exchange of data on the Web and elsewhere. The two main features of XML are that the data is organized hierarchically, and is semi-structured, mixing content, e.g., text and structure, using so called XML tags. A file conforming to the XML format is called an XML document. To constrain the structure and content of an XML document, the W3C has proposed a schema language, XML Schema. XML Schema allows to define which tags are used to define XML elements, the basic hierarchical structure of an XML document, and what data types values comply to. It is often the case that not all information about a candidate is useable or useful for duplicate detection. For instance, a candidate of type movie might be effectively described using information about its title, director, production year, etc. However, the textual review of a movie is usually not a useful indicator. In principle, one could choose any data item as part of the description of a candidate. In practice, a candidate description comprises sibling, ancestor, or descendent data, such as attribute values of a tuple, or children of an XML element. Due to the hierarchical structure of XML, elements relate to their ancestors and descendants, and may further relate to elements anywhere else in the XML document, e.g., through keyref elements. When considering duplicate detection for elements of different types, as we do, these relationships can be considered to improve duplicate detection. To detect duplicates among candidate elements, we do not want to consider unnecessary information. On schema level, we have defined a candidate type description that restricts information relevant for duplicate detection for every candidate type **tcand** $\in$ Tcand to a set of relevant element types **CTDE(tcand)** and a set of relevant and related candidates **CTDR(tcand)**. Based on these definitions on schema level, we define the candidate description **CD(c)** of a candidate of type tcand to include both the element instances defined by **CTDE(tcand)** and the related candidates according to **CTDR(tcand).** The set of element instance describing

a candidate c, defined by **CTDE(tcand)** is called element-based candidate description **CDE(c)**.

II. Iterative Duplicate Detection

The approach that was selected for duplicate detection is an iterative approach that compares candidate pairwise pairs using a similarity measure. More specifically, a similarity score between two distinct candidates c and c′ of same candidate type tcand is computed, and if the similarity is above a given threshold, c and c′ are duplicates, otherwise they are considered to be non-duplicates. The result of iterative duplicate detection is a set of duplicate candidate pairs. Iterative duplicate detection returns pairs of duplicate candidates. However, a candidate can be a member of several duplicate pairs, and we are interested in all representations of a same real-world object, not only pairs of representations. Due to the transitivity of the "is-duplicate-of" relationship, it is true that if candidate c1 is duplicate of c2 and c2 is duplicate of c3, then c1 is also duplicate of c3. To obtain the final result that consists of clusters of candidates, the transitive closure can be applied over duplicate pairs to obtain clusters with more than one candidate. Candidates not having any duplicate do not appear in the set of duplicate pairs. These can be added to the final result as clusters containing only one candidate.

The effectiveness of duplicate detection describes the quality of the duplicate detection result. Generally, it considers both (i) the set of false positives, i.e., pairs classified as duplicates that are actually not duplicates, and (ii) the set of false negatives, i.e., pairs that were not classified as duplicates, although they are. The lower the number of false positives and false negatives, the better the effectiveness. Typical metrics to measure the effectiveness of a duplicate detection approach are recall, precision, and their harmonic mean, i.e., the f-measure. The efficiency of a duplicate detection approach is defined by an analysis or measure of its computational/time complexity. The lower its time complexity, the more efficient a duplicate detection approach is. Analysis consists of a theoretical time complexity analysis including the best, worst, or average case of duplicate detection. To measure the time complexity of an algorithm, we measure both the number of pairwise comparisons and the time needed to perform all comparisons. The scalability of a duplicate detection approach is defined by an analysis or measure of its space complexity. The lower its space complexity, the more scalable a duplicate detection approach is. To assess the scalability of an approach, we apply duplicate detection algorithms to large amounts of data not fitting in main memory, and measure the corresponding runtime. XML duplicate detection requires the detection of duplicates among multiple candidate types, a problem that has also been considered recently for relational data. In both data models, it is possible and sensible to distinguish between candidate types, element based candidate type descriptions, and relationship-based candidate type descriptions.

However, it is more challenging to distinguish these three types of information in XML than in relational data. In relational schemas, a candidate type corresponds to a relational table and relationship-based candidate type descriptions are given by foreign keys. Attributes of a table compose the element-based candidate type description of the candidate the table describes. On instance level, a tuple is considered as a candidate. Its relationship-based description consists of the set of tuples referencing it by foreign key, and its element-based description is the set of attribute value pairs of that tuple. Opposed to the clear distinction between candidate types and descriptions in relational schemas, XML schemas represent all information within nested XML elements. Nesting, and possibly keyref constraints can be used to determine the relationship-based candidate type description of a candidate type. However, there is no clear structural distinction between candidates and their element-based description, because both are represented by XML elements. Consequently, the task of determining the element-based candidate description for a candidate c.

III. Conclusion

In this paper, it is attempted to look into the key aspects of duplicate detection using iterative method. It is understood that XML duplicate detection involves detection of duplicates among multiple candidate types.

References

[1] R. Ananthakrishna, S. Chaudhuri, V. Ganti,"Eliminating fuzzy duplicates in data warehouses", In International Conference on Very Large Databases (VLDB), Hong Kong, China, 2002.

[2] R. A. Baeza-Yates, B. A. Ribeiro-Neto,"Modern information retrieval", ACM Press / Addison-Wesley, 1999.

[3] P. Bertolazzi, L. D. Santis, M. Scannapieco,"Automatic record matching in cooperative information systems", In Workshop on Data Quality in Cooperative Information Systems, pp. 13–20, Siena, Italy, January 2003.

[4] I. Bhattacharya, L. Getoor,"Iterative record linkage for cleaning and integration", SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD)", 2004.

[5] I. Bhattacharya, L. Getoor,"Relational clustering for multi-type entity resolution", Workshop on Multi-Relational Data Mining (MRDM), 2005.

[6] T. Böhme, E. Rahm,"XMach-1: A benchmark for XML data management", In Proceedings of BTW 2001, pp. 264–273, Oldenburg, Germany, March 2001.

[7] M. Bilenko, R. Mooney,"On evaluation and training-set construction for duplicate detection", In Proceedings of the KDD 2003 Workshop on Data Cleaning, Record Linkage, and Object Consolidation, pp. 7–12, 2003.

Mrs.Ch. Sita Kameswari, M.C.A, M.Tech., M.B.A (Ph.D) is M.Tech (co-ordinator) BABA Instititute of Technology & Sciences, A Lady of true vision towards modren professional education and deep routed values.

Mrs. Ch. Sita Kameswari has come in association with BITS in the capacity of Professor & HOD, Dept. of CSE and PG Coordinator. She Pursued her MCA in the year 1997 and M.Tech (CSE) with specialization in Artificial Intelligence and Robotics from Andhra University in the year 2008. Her quest for knowledge led her to attain the MBA degree from IGNOU. Currently, she is at the verge of submission of her thesis with GITAM university for the award of Ph.D.

She has a reckonable and Meritorious experience of 15 years in the teaching field since 1997. During this period she served for various professional colleges like J.A. Karia College, Jamnagar, Gujarat; Boston college for professional studies, Gwalior; ANITS, Visakhapatnam etc. in the capacity of Asst Professor, Associate Professor and also discharged the additional responsibilities as HOD of MCA and CSE. She was ratified for the post Associate Professor by AU, Visakhapatnam nad RGPV, Bhopal.

She has vast experience in the administration of departments, facilitating internships, project guidance and counseling of students on the progressive path. She played an active role as management representative for ISO 9001 certification during her tenure at Boston College Gwalior.

She had Organized a joint International Conference on Swarm, Evolutionary and Memetic Computing (SEMCCO) and Fuzzy and Neural Computing Confrence (FANCCO) at ANITS, Visakhapatnam Successfully in the position of organizing secretary in the month of Dec 2011 she organized I CSI AP sytudent convention at ANITS, in coordination with CSI successfully as co-convener she organized various National level student Fests such as TECHNOCOM 2K6, AADYOTA-08, AADYOTA-09 in the institutions she worked with and earned the appreciation of one and all.

She had published her research papers in 2 International Journals, 2 proceedings of International Conference and 2 National Conferences. She also presented papers in International and National Conferences. A few more papers of her are under processing for publication.

She actively participated in about 20 workshopes organized by Indian Science Congress and other Professional bodies at various organizations. Her areas of interest are Artificial Intelligence, Computer Graphics, Object Oriented Software Engineering, Operating Systems, System Programming, Machine Learning, Neural Networks.

Her hobbies include Listening to old and new melodies, reading books and playing shuttle badminton.

She believes in the wording of swami Vivekananda "Arise, Awake and Stop not till the Goal is Reached".

Mrs P.K.R. Madhuri, Student of M.Tech in Computer Science & Technology from BABA Institute of Technology & Sciences, Visakhapatnam, AP, India.