XML SIMILARITY EVALUATION USING STRUCTURAL AND SEMANTIC SIMILARITY

P.Selvakumar[1]
[1]PG Scholar, K.S.Rangasamy College of Technology, Tiruchengode, India.
Email: selvacse.ksr@gmail.com, Mobile No: +91 9952756626.

***Abstract*- XML similarity evaluation has become a vital role in the database and information communities. XML similarity evaluation applications are clustering of document, control of version, integration of data and ranked retrieval. Various algorithms for comparing hierarchically structured data and particularly for XML documents have been proposed in the literature. The techniques for finding the edit distance between tree structures, XML documents being commonly modeled as Ordered Labeled Trees. Xml similarity evaluation using structural and semantic similarity provide an integrated and fine-grained comparison framework to deal with both structural and semantic similarities in XML documents and to allow the end-user to adjust the comparison process according to requirements. The investigation of current approaches lead to find several similarity aspects such as sub-tree related structural and semantic similarities, which are not sufficiently addressed while detecting the occurrences and repetitions of structurally and semantically similar sub-trees.**

***Keywords*-XML, Ordered Label Trees, semantic similarities, sub trees.**

I. INTRODUCTION

The Extensible Mark-up Language is seeing increased use, and in the future also the application is used more. But many of these XML documents, especially those beginning to appear on the web, are without Document Type Descriptors (DTDs). It provides a method to automatically extract a DTD for a set of XML documents. Several benefits for the existence of DTDs are provided by them. The use of XML covers data description and storage, database information interchange, data filtering, as well as web services interaction. The increasing web exploitation of XML, XML document comparison becomes a central issue in the database and information retrieval communities. Due to the growth of the World Wide Web, there is an increasing need to automatically process Web documents for efficient data management, similarity clustering and search applications. While HTML (Hyper Text Markup Language) provides a rather visual markup, having knowledge of the logical structure of the data is a fundamental prerequisite for the interoperability of web based information systems. XML was introduced by the W3C as an efficient means for data representation and management. The main goal of the paper is the comparison of rigorously structured heterogeneous XML documents, i.e., documents originating from different data-sources and not conforming to the same grammar (DTD/XSD), which in case a lot of XML documents found on the Web.

A. Motivations:

To provide a fine-grained method that captures both structural and semantic similarities when comparing XML document structures is the main objective of the paper. Here the motivations of the work, highlighting the relevance of structural and semantic similarity evaluation in XML document comparison. Here specifically focus on similarities left unaddressed in current approaches, which aims to capture with our XML document similarity measure.

B. Proposal:

The problem of XML document structure comparison is that the detecting of the occurrences and repetitions of structurally/semantically similar sub-trees. In sub-trees, just underline structures made of multiple nodes, as well as single leaf nodes. So the aim to provide a unified and fine-grained method to deal with both structural and semantic resemblances left addressed by existing comparison methods. The XML comparison method consists of four main algorithms:

i. Struct_CBS for identifying the Structural Commonality Between two Sub-trees,
ii. Sem_RBS for quantifying the Semantic Resemblance Between two Sub-trees,
iii. TOCXDoc for computing the Tree edit distance Operations Costs,
iv. TEDXDoc for computing the Tree Edit Distance between XML document trees.

In short, the TOC algorithm makes use of Struct_CBS and Sem_RBS to structurally and semantically compare all sub-trees in the XML documents being compared. The produced sub-tree similarity results are consequently exploited as edit operations costs (particularly tree insertion and tree deletion costs, which are central to detecting the occurrences and repetitions of similar sub-trees), in an adapted version of main edit distance algorithm, which we identify as TED. Hence, the inputs to our XML comparison approach are as follows:

- the XML document trees to be compared,
- parameter a enabling the user to assign more importance to the structural or semantic aspects of the XML documents being treated,
- a reference (weighted) semantic network SN, for semantic similarity evaluation.

Consequently, the method outputs the similarity between the XML document trees being compared.

II. STRUCTURAL SIMILARITY BETWEEN SUB-TREES (STRUCT-CBS)

Sub-tree structural similarities are usually left undetected in current XML comparison approaches. The structural similaritiy method is used to Comparing the structure of two XML documents. LD(Leave-Distance) representation as input and produces the Normalized commonality between two representation. If two sub-trees are identical then the normalized value should be one. If two sub-trees have no relation then the normalized value must be zero. Otherwise between 0 to 1 the Struct_CBS algorithm, based on the edit distance concept, to identify the structural commonality between sub-trees In Struct_CBS, sub-trees are treated in their ld-pair representations. Using the ldpair tree representations, sub-trees are transformed into modified sequences (ld-pairs), making them suitable for standard edit distance computations. The algorithm starts by computing the sum of the costs of deleting every node in the source sub-tree and inserting every node of the destination tree. Consequently, it identifies the set of insertion/deletion operations having the minimum overall cost Structurally matching nodes are associated null costs. Note that the update operation is specifically disregarded in Struct-CBS, in order to allow the identification of structurally matching nodes. Consequently, the overall sum of the minimum operations' costs underlines an edit distance, i.e., Dist[first sub tree][second sub tree], between the sub-trees first and second being compared. Hence, the maximum number of matching nodes between first and second, StructCom (first, second).

Dist[n][m] = min{ Dist[n-1][m-1] ,Dist[n-1][m] + CostDel(SbTi[n]),
Dist[n][m-1] +CostIns(SbTj[m])
Return
| SbT i|+| SbTj| - Dist[|SbTi |][|SbTj |]/2*(Max(|SbTi|,|SbTj|))

Therefore finding the normalized commonality is

First sub tree + second sub tree – distance[first sub tree][second sub tree]/ 2*max[first sub tree,second sub tree].

III. SEMANTIC RESEMBLANCE BETWEEN SUB-TREES

Comparing the semantic characteristic of two XML documents. The following similarities can also identified. Occurrence of semantically similar sub-trees Semantic similaritybetween a sub-trees occurring at different depths. Semantic similarity between sub-tree and whole XML Tree

Dist[n][m] = min{ If (SbTi[n].d = SbTj[m].d & SbTi[n].l = SbTj[m].l) {Dist[n-1][m-1] },Dist[n-1][m] + CostDel(SbTi[n]), Dist[n][m-1] +CostIns(SbTj[m])}}

Return

| SbT i|+| SbTj| - Dist[|SbTi |][|SbTj |]/2*(Max(|SbTi|,|SbTj|))

IV. TREE EDIT OPERATIONS COSTS (TOC)

Finding operation cost of each sub-tree edit operation. It have the XML documents as input and produces the structural & semantic similarity, weighted semantic network. The minimum cost for inserting/deleting a single node sub-tree is equal to 0.5,half its maximum insert/delete cost.

Tree edit operations costs **(**TOC), the similarity between two XML sub-trees, SS(first,second) is evaluated as the weighted average of their structural commonality and semantic resemblance.

Ss(first sub tree, second sub tree,α)=α*struct_cbs(first sub tree, second sub tree)+(1-α)*sem_RBS(first sub tree, second sub tree).

The user can thus assign more importance to either structural or semantic similarities by varying parameter is [0,1]

- For a = 1, TOC will only consider structural commonalities in computing operations costs (via Struct_CBS).
- For a = 0, only sub-tree semantic resemblances will be considered in computing operations costs (via Sem_RBS).

V. TREE EDIT DISTANCE (TED)

Finding the edit distance of each sub-tree in XML documents. It have XML document as input and produces sub-tree insertion/deletion operation cost, structural/semantic weighting, weighted semantic network as output. Maximum cost when the compared element labels are identical is zero. Otherwise the maximum unit cost is one.

VI. OVERALL COMPLEXITY

A. Time Complexity

The overall complexity of our integrated structural and semantic similarity approach simplifies to O(|A|X |B|X |SN| X Depth(SN)), where |A| and |B| denote the cardinalities of the compared trees, |SN| the cardinality of the semantic network exploited for semantic similarity assessment, and Depth(SN) its maximum depth. It is computed as follows

- Struct_CBS algorithm for the identification of the structural commonality between two sub-trees is of complexity: O(|SbT_i| X |SbT_j|) where |SbT_i| and |SbT_j| denote the cardinalities of the compared sub-trees.
- Sem_RBSfor identifying the semantic resemblance between two sub-trees is of complexity: O(|SbT_i| X |SbT_j| X |SN| X Depth(SN)). Note that O(|SN| X Depth(SN)) underlines the time complexity of the semantic similarity measure.

B. Space Complexity

As for memory usage, our approach requires RAM space to store the XML document trees being compared, as well as the distance matrixes and semantic vectors being computed. It simplifies to O(|A| X |B|) space.

- Struct_CBS requires |SbT_i| X |SbT_j| space for storing the distance matrix when identifying the structural commonalities between any two sub-trees SbT_i and SbT_j. Hence, space complexity is of O(|SbT_i| X |SbT_j|).
- Sem_RBS requires 2 X (|SbT_i| + |SbT_j|) space for handling corresponding sub-tree vectors, each vector being of maximal dimension |SbT_i| + |SbT_j|. Hence, Sem_RBS is of O(|SbT_i| + |SbT_j|).Note that the semantic network is not stored in local memory, but is stored on disk and thus does not contribute to space complexity.

VII. CONCLUSION

The paper consists of fine-grained similarity approach for comparing rigorously structured XML documents. The target document structure (i.e., structure-only XML, consisting of element/attribute tag names) and disregard content (i.e., element/ attribute values), central in structural clustering/classification and structural querying applications. Here the method combines tree edit distance computations and information retrieval semantic similarity assessment, so as to capture the structural and semantic resemblances between XML documents. Particularly focus on previously unaddressed sub-tree structural and semantic similarities, allowing the user to tune the comparison process according to her requirements and needs. The theoretical study and experimental evaluation showed that the approach yields improved similarity results with respect to the existing alternatives. Timing analysis underlined the impact of semantic similarity assessment, due to traversing the semantic network at hand.

REFERENCES

[1] A.Formica, M.Missikoff , “Concept similarity in SymOntos: an enterprise ontology management tool” 2002.

[2] A.Nierman, H.V.Jagadish, “Evaluating structural similarity in XML documents” 2002.

[3] Ales Wojnar, Irena Mly nkova, Jiri Dokulil, “Structural and semantic aspects of similarity of Document Type Definitions.

[4] Alsayed Algergawy, Eike Schallehn, Gunter Saake, “Improving XML schema matching performance using Prüfer sequences” 2009.

[5] Alsayed Algergawy, Richi Nayak, Gunter Saake, “Element similarity measures in XML schema matching” 2010.

[6] Buhwan Jeong, Daewon Lee, Hyunbo Cho, Jaewook Lee, "A novel method for measuring semantic similarity for XML schema matching" 2007.

[7] D.Rafiei, "Finding syntactic similarities between XML documents" 2006.

[8] J.Tekli, R. Chbeir, K. Yetongnon, "A fine-grained XML structural comparison approach" 2007.

[9] Jesús Oliva, Jose Ignacio Serrano, Maria Dolores Del Castillo, Angel Iglesias, "SyMSS: A syntax-based measure for short-text semantic similarity" 2011.

[10] Joe Tekli, Richard Chbeir, "A novel XML document structure comparison framework based-on sub-tree commonalities and label semantics" 2011.

[11] Joe Tekli, Richard Chbeir, Kokou Yetongnon, "Semantic and Structure Based XML Similarity: The XS3 Prototype" 2006.

[12] R. Schenkel, "Semantic similarity search on semi structured data with the XXL search engine" 2005.

[13] S.Helmer, "Measuring the structural similarity of semi structured documents using entropy" 2007.

[14] T.Schlieder, "Similarity search in XML data using cost-based query Transformations" 2001.

[15] Wang Tiantian, Su Xiaohong, Wang Yuying, Ma Peijun, "Semantic similarity-based grading of student programs" 2006.