Structural and semantic similarity for XML comparison

Renato Guzman, Irvin Dongo, Regina Ticona Herrera

Structural and semantic similarity for XML comparison

Renato Guzman
San Pablo Catholic University
Arequipa, Peru
renato.guzman@ucsp.edu.pe

Irvin Dongo
San Pablo Catholic University
Arequipa, Peru
irvin.dongo@ucsp.edu.pe

Regina Ticona Herrera
University of Pau and Pays de l'Adour
Bayonne, France
rticona@iutbayonne.univ-pau.fr
San Pablo Catholic University
Arequipa, Peru
rticona@ucsp.edu.pe

ABSTRACT

XML has experimented a rapid growth mostly because of its application on the Web. Application varies from version control management, data storage to clustering and information retrieval. In this context, it is necessary to develop efficient techniques for comparing XML documents. Many method proposed are based only on structural commonalities, ignoring semantics. In this paper, we propose a new method for comparing XML documents based on LevelEdge combining tag structural and semantic similarities.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*search process*

General Terms

Algorithms, Measurement, Performance, Design, Experimentation.

Keywords

XML, XEdge, Level Structure, Level Edge, Structural Similarity, Semantic Similarity.

1. INTRODUCTION

XML (eXtensible Markup Language) is a markup language presented by the W3C (World Wide Web Consortium) that allows to represent hierarchically data with tags. This representation has grown as many applications on the Web have adopted this language. Due to the large amount of information represented in XML, it is needed to be managed efficiently mainly for storage and information retrieval[8]. XML document comparison is applied to several fields including versioning, classification and clustering among others.

MEDES'13 October 29-31, 2013, Neumünster Abbey, Luxembourg

Some information represented by this language has been structured using standards, but as the Internet grows there is a great diversity of heterogeneous documents that may present similar data. Structure-only methods ignore semantics and do not take advantage of the XML hierarchical scheme. In this context, structural similarity is not the only measurement that must be taken while comparing XML documents, but also semantic similarity between concepts.

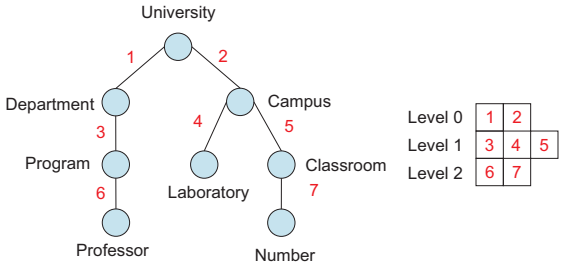
1.1 Motivation

Consider, for example, XML trees in Figures 1a and 1b. We can realize that both examples have the same context but using different words for express the nodes. We can conclude at first appearance that both have a high similarity including when they use distinct words concerning at the same context. Nonetheless, such semantic similarities are left unaddressed by existing approaches based on structural similarities as [2][1]. Moreover, the relation between the nodes gives more meaning to the context of the document and permits to develop the semantic in a complex structure (parent/child). This is another situation that the structural-grammar approaches, [7] for example does not take into account.

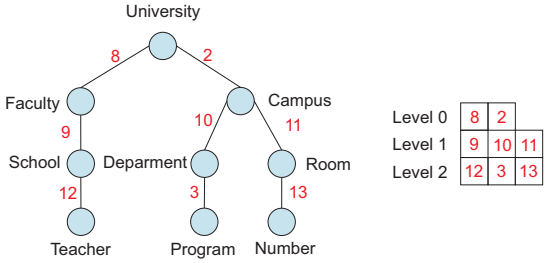
1.2 Contribution and Organization of the Paper

The goal of our paper is to provide an improved XML structural and semantic similarity method for comparing heterogeneous XML documents based on the algorithm of XEdge [1] and using the potentiality of semantics into the structure. In short, we also aim to test on existing approaches mainly [2], [7], in order to take into account the various approaches to compute commonalities while comparing XML trees.

The contribution of this study can be summarized as follows: i) introducing an approach for computing XML similarity taking into account the structure, the relation between the nodes and semantic of the nodes, ii)comparing with other approaches based on structure-only, structure-grammar to develop the difference when we introduce the semantic into the structure. The remainder of this paper is organized as follows. Section 2 reviews the background in Level Structure and Level Edge Structure. In Section 3 we develop our XML structural and semantic similarity approach with two examples using XEdge and our technique. Section 4 presents experimental results. Conclusions are covered in Section 5.



(a) Example 1



(b) Example 2

Figure 1: Level edge representation examples

2. RELATED WORK

2.1 LevelStructure

LevelStructure [4] is a compact representation of XML documents. Level structure requires to represent each XML document as an ordered labeled tree (OLT). OLTs are presented as a list of ld-pairs where each node is a tuple of its label (l) and its depth (d).

LevelStructure groups distinct tag nodes from each level and presents it as a list of levels where each level has a list of unique tags. Repeated nodes are taken into account once.

Information presented in LevelStructure are the nodes for each level in each document. Relations between these nodes are omitted. Therefore it is possible to find XML documents with different relations between nodes with the same LevelStructure representation. This can be common in XML documents derived from the same DTD where they share a strong structure similarity.

2.2 LevelEdge Structure

LevelEdge structure is introduced by [4] as a compact XML summary of the document based on edges that represent the relationship between two nodes (parent/child). The XML document is represented grouping each level as a vector of distinct edges. This representation preserves the relationship between nodes through the structure.

2.2.1 *Representation*

The LevelEdge representation is an improvement over LevelStructure, introduced by the same authors[4], because it keeps the relation between two consecutive levels. In LevelStructure representation, the XML document is summarized in vectors of distinct nodes for each level but the context of each node is lost. The main advantage of LevelEdge over LevelStructure is the preservation of structural relation between two consecutive nodes.

XML documents that are derived from different DTDs can have similar tags in different levels. With LevelStructure both documents can be considered highly similar because of the coincidence of several node tags. LevelEdge distinguishes both documents if the node tags do not share the same relation between similar node tags. XML documents that are derived from the same DTD may share sets of node tags that are similar or equal in each level. In this case LevelStructure will not differentiate such documents. With LevelEdge the documents may share most of the node tags but the relation between them may differ and therefore the level of similarity is more accurate.

LevelEdge can distinguish documents based on edges in the same level. However, even if two documents share the same set of edges in all levels, there is a chance that those documents will not be the same because the vectors of edges that represent the XML documents will have distinct edges in each level. If there is more than one edge with the same parent and child node tag, then it will be stored as a single edge in that level.

2.2.2 *Distance metric in LevelEdge Structures*

The authors in [1] propose a distance metric to measure similarity between two XML documents in LevelEdge representations. This metric compares identical edges in order to compute the final similarity.

Consider L_1 and L_2 as two LevelEdge representations of two XML documents. The similarity measure is as follows:

$$Sim_{L_1,L_2} = \frac{\sum_{i=0}^{m-1} c_i \times a^{m-i-1}}{\sum_{i=0}^{M-1} t_j \times a^{M-i-1}} \quad (1)$$

where

a: is a positive integer.

m is the minimum number of levels of L_1 and L_2.

M is the maximum number of levels of L_1 and L_2.

c_i is the number of identical edges in L_i.

t_i is the total number of edges in L_1 and L_2.

The positive integer a is a factor that measures the importance of higher levels of the representation.

Heterogeneous documents do not derive from the same DTDs therefore edges may not occur on the same levels and tags might vary even if they are semantically equivalent. The distance measure is modified as follows:

$$SimL_L = 0.5 \times \sum_{i=0}^{L-1} c_i \times a^{L-i-1} \quad (2)$$

$$Sim_{L_1,L_2} = \frac{SimL_{L_1} + SimL_{L_2}}{\sum_{j=0}^{M-1} t_j \times a^{M-j-1}} \quad (3)$$

where c_i^1 is the number of common edges found at the level i of L and the other lower levels of the LevelStructure being compared.

The procedure of this equation of two XML documents represented as L_1 and L_2 is as follows:

1. Start at level 0 of both documents. Set c_i as the number of common edges between L_1 and L_2. If there are common edges continue with Step 2, otherwise continue with Step 3.

2. Move L_1 and L_2 to the next level. Set c_i as the number of common edges between L_1 and L_2 in level i. If there are common edges continue with Step 2, otherwise continue with Step 3.

3. Move L_2 to the next level, maintaining L_1 in the same level. Set c_i as the number of common edges between L_1 and L_2 in their corresponding levels. If there are common edges continue with Step 2, otherwise continue with Step 3.

4. Repeat the procedure until all levels are checked.

Note that when similarity is 1, this does not mean that the two documents are exactly the same, but that all their edges are common in both documents. Finally the final distance between two documents is computed as follows:

$$Dist_{A,B} = 1 - Sim_{L_1,L_2} \tag{4}$$

LevelEdge representation allows to maintain relations between tags and their parents. The first distance measure works with homogeneous documents as they derive from the same DTDs, but XML has been widely used in different scenarios and therefore they may have different sources. The second distance metric works well for heterogenous documents, but it computes a comparison of identical edges, however an edge can be structurally different but semantically equivalent.

3. PROPOSAL

In this approach, the semantically differences between tags and the distance while comparing levels are taken into account. In the previous approach, the distance between the levels on node tags have no effect in the similarity measure, that is, edges being in different levels in LevelEdge representation will be treated as if they were in the same level. The a factor only measures the deepness of the edge being compared against all the edges that are in lower levels in the other LevelEdge representation. The edge being compared in the first LeveEdge representation will be affected by the a factor but the similarity between this edge and the second LevelEdge representation will not be affected.

In this approach, we introduce a modified d_factor presented in [6] which is used when comparing edges from different levels. Let L_1 and L_2 be two representations in LevelEdge of two documents and let c_i and c_j be two edges. The modified d_factor is as follows:

$$d\_factor'(c_i, c_j) = \frac{1}{1 + abs(level_{c_i} - level_{c_j})} \tag{5}$$

The d_factor component measures the distance between two edges. As more levels are between two edges, there will be less structural similarity. A $d\_factor' = 1$ will indicate that both edges share the same level in the different LevelEdge representations. Edges from the same levels in both LevelEdge representations will not be affected by this factor.

LevelEdge compares common edges in different levels structurally, that is, if they share the same node tag then it will be considered as a common edge, but even if two edges are structurally different, they can be semantically more similar than others. This approach takes into account semantic similarity while comparing two edges. Lin similarity[3] is used to measure the distance between two concepts, in this context it will be used to compare node tags from two edges. In this paper, Lin semantic similarity is based on *Wordnet*. $S(c_1, c_2)$ is defined as the semantic similarity between two edges based on Lin.

$$S(c_i, c_j) = \frac{sim_{lin}(c_{i_1}, c_{j_1}) + sim_{lin}(c_{i_2}, c_{j_2})}{2} \tag{6}$$

where c_{i_1} and c_{j_1} are the parents of both edges and c_{i_2} and c_{j_2} the children of c_i and c_j respectively. $S(c_i, c_j)$ will be 1 if the parent node tags in both LevelEdge structures have the same or equivalent semantic similarity and also the child node tags have the same or equivalent semantic similarity, in this case the edges being compared will be semantically equivalent. This semantic similarity will be affected by the d_factor if they are in different levels.

The similarity between two edges based on d_factor' and S values is given by:

$$sim_c(c_i, c_j) = d\_factor'(c_i, c_j) \times S(c_i, c_j) \tag{7}$$

This similarity takes into account the structure distance between two edges and the semantic similarity between parent and child node tags. With this similarity, we find the similarity value for each edge and the other LevelEdge representation using the following formula:

$$sim_{level}(c_i, L) = \sum_{j=i}^{|L|} \sum_{k=0}^{|L_j|} sim_c(c_i, L_{j_k}) \tag{8}$$

where $|L|$ represents the number of levels of the LevelEdge representation L. The positive integer a remains as in the initial distance measure of LevelEdge representation.

Given two LevelEdge representations L_1 and L_2, let c_i be an edge at level i from L_1 compared to L_2. The procedure for the previous equation is as follows:

1. Start by positioning L at level i. The vector of edges at level i will be L_j.

2. For each edge in L_j calculate the similarity measure between c_i and L_{j_k}. The final similarity will be modified by the influence of the level.

3. Repeat until there is no more level at L.

$SimL'$ represents the similarity between each edge from each level from LevelEdge representation L and another LevelEdge representation L'. $SimL'$ is defined as follows:

$$SimL'_{L,L'} = 0.5 \times \sum_{i=0}^{|L|} (\sum_{j=0}^{|L_i|} sim_{level}(c_i, L')) \times a^{|L|-i} \tag{9}$$

The final similarity between two level structures is as follows:

$$Sim'(L_1, L_2) = \frac{SimL'_{L_1,L_2} + SimL'_{L_2,L_1}}{\sum_{i=0}^{M} n_i \times a^{M-1}} \tag{10}$$

where M represents the total number of distinct edges in L_1 and L_2 and n_i is the sum of edges that have been compared at each level.

This similarity measure has a value between 0 and 1. A value of 0 represents two different documents and a value of 1 represents documents with high similarity.

In order to illustrate our approach and how it compares to LevelEdge similarity algorithm, we present a computation example for both algorithms. In these examples, L_1 and L_2 will be the representions of LevelEdge structures from the figure 1a and 1b respectively.

3.1 LevelEdge Computation example

Figure 1 presents two representations of XML documents and its corresponding LevelEdge representation, both representations share common node tags and a high semantic similarity.

The similarity between two the LevelEdge representation from figure 1 based on the equation 3 is:

$$Sim'L_{L_1} = 0.5 \times (1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0) = 3.0$$

$$Sim'L_{L_2} = 0.5 \times (1 \times 2^2 + 0 \times 2_1 + 1 \times 2_0) = 2.5$$

$$Sim'_{L_1,L_2} = \frac{3.0 + 2.5}{3 \times 2^2 + 6 \times 2^1 + 5 \times 2^0} = 0,1897$$

The final similarity is low between these representations because they share few equal nodes between their levels.

3.2 Computation example Our approach

Consider the two XML representations from figure 1. Due to space limitation, there will be compared only an example for each equation from our approach.

The first step is to compute sim_c between two edges from the LevelEdge representation from the equation 7. We will consider the edges numbered 1 and 9 which corresponds to the edges *(university, department)* and *(faculty, school)* respectively. In order to get the sim_c, the S and d_factor values need to be calculated as follows:

$$S(1,9) = \frac{lin(university, faculty) + lin(department, school)}{2}$$

$$S(1,9) = \frac{0.1015 + 0.5590}{2} = 0.3302$$

$$d\_factor'(1,9) = \frac{1}{1 + abs(0-1)} = 0.5$$

The final value for sim_c between these two edges is:

$$sim_c(1,9) = d\_factor(1,9) \times S(1,9) = 0.1651$$

Considering that we have computed all the sim_c values required for the computation of equation 8 sim_{level} that corresponds to the similarity between the edge and the LevelEdge structure being compared, the sim_{level} value is:

$$sim_{level}(1, L_2) = 1.6574$$

Method	Precision	Recall	F-Value
Chawathe	0.44	0.4314	0.4356
Tekli	0.56	0.549	0.5545
XEdge	0.9875	0.9875	0.9875
New	0.995	0.995	0.995

Table 1: precision, recall and F-Value

The value of the constant $a = 2$ is considered as it is in [1], the optimal value for this factor will not be treated in this paper.

The similarity sim_L from equation 9 that corresponds to the partial similarity between the first LevelEdge structure and the second LevelEdge structure is:

$$SimL'_{L_1,L_2} = 59.5552$$

$$SimL'_{L_2,L_1} = 47.8496$$

The final similarity taken into account the values from sim_L for both LevelEdge structures will be:

$$Sim'_{L_1,L_2} = \frac{59.5552 + 47.8496}{(13+14) \times 2^2 + (11+12) \times 2^1 + 5 \times 20}$$

$$Sim'_{L_1,L_2} = 0.7726$$

The final similarity is higher than the LevelEdge structures beacuse it considers semantic similarity in edges.

4. EXPERIMENTAL RESULTS

Experimental results were performed with a dataset of 200 documents derived from 8 DTDs extracted from Wisconsin-Madison University [1]. The documents have a total of 56,971 nodes, an average depth of 3.965 and an average of 284.85 nodes per document.

Single Hierarchical Clustering[5] was used in order to classify the documents to its corresponding DTDs to test the effectiveness of this new method and we applied three measures: precision, recall and F-value.

Table 1 shows the results for the test and a comparison with the algorithms [2], [1], [7] and the proposed approach. Our approach performed better than the other algorithms and improved the performance of its predecessor XEdge. Chawathe is a structure only algorithm based on edit distance. Chawathe did not perform well as it only matches equal node tags and does not take into account semantic similarity. Tekli introduces semantic similarity and it is also a edit distance algorithm. Both algorithms performs comparison based on a single node, XEdge extends the comparison to edges in the structures which increased the values measured in the tests. Our approach in addition to XEdge introduces semantics and level differences which had a positive impact on the results. Our algorithm and XEdge classified almost all the documents on its correct DTD grammar.

5. CONCLUSIONS

In this paper a new similarity measure was presented to compare XML documents. The documents were represented

[1] http://research.cs.wisc.edu/niagara/data.html

in LevelEdge Structure and the similarity measure was based on XEdge distance metric. The new method compared structural and semantical characteristics and provided a better accuracy in heterogeneous documents. The introduction of semantics in heterogeneous documents in the LevelEdge representation improved the original XEdge algorithm and also the other structure-only and semantic algorithms.

6. REFERENCES

[1] P. Antonellis, C. Makris, and N. Tsirakis. Xedge: clustering homogeneous and heterogeneous xml documents using edge summaries. In *Proceedings of the 2008 ACM symposium on Applied computing*, SAC '08, pages 1081–1088, New York, NY, USA, 2008. ACM.

[2] S. S. Chawathe. Comparing hierarchical data in external memory. In *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, pages 90–101, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[3] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[4] R. Nayak and S. Xu. Xcls: A fast and effective clustering algorithm for heterogenous xml documents. *Lecture Notes in Computer Science*, pages 292–302, 2006.

[5] R. Sibson. Slink: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.

[6] J. Tekli and R. Chbeir. A novel xml document structure comparison framework based-on sub-tree commonalities and label semantics. *Web Semant.*, 11:14–40, Mar. 2012.

[7] J. Tekli, R. Chbeir, and K. Yetongnon. Structural similarity evaluation between xml documents and dtds. In *Proceedings of the 8th international conference on Web information systems engineering*, WISE'07, pages 196–211, Berlin, Heidelberg, 2007. Springer-Verlag.

[8] Q. Wang, Z. Ren, L. Dong, and Z. Sheng. Path-based xml relational storage approach. *Physics Procedia*, 33(0):1621 – 1625, 2012. 2012 International Conference on Medical Physics and Biomedical Engineering (ICMPBE2012).