Semantic and Structure Based XML Similarity: 
An Integrated Approach

Joe Tekli  
LE2I Laboratory CNRS  
University of Bourgogne  
21078 Dijon Cedex  
France  
joe.tekli@khali.u-bourgogne.fr

Richard Chbeir  
LE2I Laboratory CNRS  
University of Bourgogne  
21078 Dijon Cedex  
France  
richard.chbeir@u-bourgogne.fr

Kokou Yetongnon  
LE2I Laboratory CNRS  
University of Bourgogne  
21078 Dijon Cedex  
France  
kokou.yetongnon@u-bourgogne.fr

Abstract
Since the last decade, XML has gained growing importance as a major means for information management, and has become inevitable for complex data representation. Due to an unprecedented increasing use of the XML standard, developing efficient techniques for comparing XML-based documents becomes crucial in information retrieval (IR) research. A range of algorithms for comparing hierarchically structured data, e.g. XML documents, have been proposed in the literature. However, to our knowledge, most of them focus exclusively on comparing documents based on structural features, overlooking the semantics involved. In this paper, we deal with this problem and introduce a combined structural/semantic XML similarity approach. Our method integrates IR semantic similarity assessment in an edit distance algorithm, seeking to amend similarity judgments when comparing XML-based documents. Different from previous works, our approach comprises of an original edit distance operation cost model, introducing semantic relatedness of XML element/attribute labels, in traditional edit distance computations. A discussion about our similarity method’s properties, chiefly symmetry and triangular inequality, with respect to existing measures in the literature is provided here. A prototype has been developed to evaluate the performance of our approach. Experimental results were noticeable.

1. Introduction
In recent years, W3C’s XML (eXtensible Mark-up Language) has been accepted as a major means for efficient data management and exchange. The use of XML ranges over information formatting and storage, database information interchange, data filtering, as well as web services interaction. Due to the ever-increasing web exploitation of XML, an efficient approach to compare XML-based documents becomes crucial in information retrieval (IR).

Notionally, an XML document should conform to a given grammar (DTD - Document Type Definition - or XML Schema), the latter defining the overall structure of the corresponding XML document (elements, associated attributes, as well as the rules to which those elements/attributes should obey in the XML document) [19]. However, XML documents published on the Web are often found without grammars, in particular those created from legacy HTML [17]. Therefore, the need to compare heterogeneous XML documents arises. This study focuses on the problem of identifying similarities between XML documents that lack DTDs/Schemas.

A range of algorithms for comparing semi-structured data, e.g. XML documents, have been proposed in the literature. All of these approaches focus exclusively on the structure of documents, ignoring the semantics involved. However, in the field of information retrieval (IR), estimating semantic similarity between web pages is of key importance to improving search results [15]. The relevance of semantic similarity in IR research, as well as the unprecedented abundant use of XML-based documents on the web, incited us to expand XML structural similarity so as to take into account semantic relatedness while comparing XML documents.
In order to stress the need for semantic relatedness assessment in XML document comparisons, consider the examples in Figure 1.

![XML tree diagram]

**Fig. 1.** Examples of XML documents

Using traditional edit distance computations, the same structural similarity value is obtained when document A is compared to documents B and C (structural similarity computations are detailed in Section 3.1.2). However, despite having similar structural characteristics, one can obviously recognize that sample document A shares more semantic characteristics with document B than with C. Pairs Academy-College and Professor-Lecturer, from documents A and B, are semantically similar while Academy-Factory and Professor-Supervisor, from documents A and C, are semantically different. It is such semantic resemblances/differences that we aim to take into consideration while estimating similarity between XML documents. In this study, we integrate semantic similarity assessment in a structured XML similarity approach, in order to provide an improved XML similarity measure for comparing heterogeneous XML documents.

The remainder of this paper is organized as follows. Section 2 briefly reviews background in both XML structural similarity approaches and IR semantic similarity methods. Section 3 develops our integrated semantic and structure based XML similarity approach. Section 4 discusses our method’s properties, mainly symmetricity and triangular inequality. Section 5 presents our prototype and experimental tests. Section 6 concludes the paper and outlines future research directions.

## 2. Background

### 2.1 XML data model

XML documents represent hierarchically structured information and can be modeled as Ordered Labeled Trees (OLTs) [27]. Nodes in a traditional DOM (Document Object Model) ordered labeled tree represent document elements and are labeled with corresponding element tag names. Element attributes mark the nodes of their containing elements. However, to incorporate attributes in their similarity computations, the authors in [17, 29] have considered OLTs with distinct attribute nodes, labeled with corresponding attribute names. Attribute nodes appear as children of their encompassing element nodes, sorted by attribute name, and appearing before all sub-element siblings [17]. In addition, in [17] and [8], both authors agree on disregarding element/attribute values while studying the structural properties of XML documents.

### 2.2 XML structural similarity

Various methods, for determining structural similarities between hierarchically structured data, particularly XML documents, have been proposed in the literature. Most of them derive, in one way or another, the dynamic programming techniques for finding edit distance between strings [12, 25]. In essence, all these approaches aim at finding the cheapest sequence of edit operations that can transform one tree into another. Nevertheless, tree edit distance algorithms can be distinguished by the set of edit operations that are allowed as well as overall complexity and performance levels.

Early approaches [28, 23] allow insertion, deletion and relabeling of nodes anywhere in the tree. However, they are relatively greedy in complexity. For instance, the approach in [23] has a time complexity $O(|A||B| \text{depth}(A) \text{depth}(B))$ when finding the minimum edit distance between two trees A and B ($|A|$ and $|B|$ denote tree cardinalities while depth(A) and depth(B) are the depths of the trees). In [4, 6], the authors restrict insertion and deletion operations to leaf nodes and add a move operator that can relocate a sub-tree, as a single edit operation, from one parent to another. However, corresponding algorithms do not guaranty optimal results. Recent work by Chawathe [5] restricts insertion and deletion operations to leaf nodes, and allows the relabeling of nodes anywhere in the tree, while disregarding the move operation. The overall complexity of Chawathe’s algorithm is of $O(N^3)$. Nierman and Jagadish [17] extend the approach provided by Chawathe in [5] by adding two new operations: insert tree and delete tree to allow insertion and deletion of whole sub-trees within an OLT. Their approach’s overall complexity simplifies to $O(N^2)$. Experimental results, given by Nierman and Jagadish [17], show that their algorithm outperforms that of Chawathe [5], which in turn yields better results than the algorithm presented in [23]. However, the authors in [17] state that their algorithm is conceptually more complex than its predecessor and that it requires a pre-computation phase, relative to determining the costs of tree insert and delete operations, which complexity is of $O(2N^2+N^3)$.

An original structural similarity approach is presented in [8]. It disregards OLTs and utilizes the Fast Fourier Transform to compute similarity between XML documents. However, the authors in [8] didn’t compare their algorithm’s optimality to existing edit distance approaches.
2.3 Semantic similarity

Measures of semantic similarity are of key importance in evaluating the effectiveness of web search mechanisms in finding and ranking results [15]. In the fields of Natural Language Processing (NLP) and Information Retrieval (IR), knowledge bases (thesauri, taxonomies and/or ontologies) provide a framework for organizing words (expressions) into a semantic space [10]. Therefore, several methods have been proposed in the literature to determine semantic similarity between concepts in a knowledge base. They can be categorized as: edge-based approaches and node-based approaches.

The edge-based approach is a natural and straightforward way to evaluate semantic similarity in a knowledge base. In [18, 11], the authors estimate the distance between nodes corresponding to the concepts being compared: the shorter the path from one node to another, the more similar they are. Nevertheless, a widely known problem with the edge-based approach is that it often relies on the notion that links in the knowledge base represent uniform distances [20, 10]. In real knowledge bases, the distance covered by a single link can vary with regard to network density, node depth and information content of corresponding nodes [21, 10]. Jiang and Conrath [10] add that link distances could also vary according to link type.

On the other hand, node-based approaches get round the problem of varying link distances. In [20], Resnick puts forward a central node-based method, where the semantic similarity between two concepts is approximated by the information content of their most specific common ancestor. Resnick’s experiments [20] show that his similarity measure is a better predictor of human word similarity ratings, in comparison with a variant of the edge counting method [18, 11]. Resnick [20] adds that his measure is not sensitive to the problem of varying distances, since it targets the information content of concepts rather than their distances from one another. Improving on Resnick’s method [20], Lin [13] presents a formal definition of the intuitive notion of similarity, and derives an information content measure from a set of predefined assumptions regarding commonalities and differences. Lin’s experiments [13] show that the latter information content measure yields higher correlation with human judgment in comparison with Resnick’s measure [20]. Furthermore, Lin’s measure is generalized by Maguitman et al. [15] to deal with ontologies of hierarchical (made by IS-A links) and non-hierarchical components (made by cross links of different types), the Lin measure (as most semantic similarity measures) targeting hierarchical structures (taxonomies).

In recent years, there have been a few attempts to integrate semantic and structural similarity in the XML comparison process. The authors in [2, 3, 22] identify the need to support tag similarity (synonyms and stems) instead of tag syntactic equality while comparing XML documents. However, the approaches in [2, 3, 22] are based on heuristic measures which disregard the edit distance computations (w.r.t. structure) and only consider the synonymy/stem relations (w.r.t. semantic similarity).

In this study, we aim to combine IR semantic similarity (taking into account the various semantic relations encompassed in the taxonomy/ontology considered in the comparison process) and an edit distance structural similarity algorithm, in order to define a semantic/structural similarity measure for comparing XML documents.

3. Proposal

Our approach consists of an original edit distance operation cost model in which semantic relatedness of XML element/attribute labels is introduced in traditional edit distance computations. In Section 3.1, we present the edit distance process utilized in our study. Section 3.2 develops our integrated semantic/structure based method.

3.1 Structural similarity

Our investigations of the various structural similarity methods proposed in the literature led us to adopt Chawathe’s approach [5], his algorithm’s performance being recognized and, therefore, further specialized by Nieman and Jagadish [17]. In addition, Chawathe’s approach [5] is a direct adaptation of Wagner and Fisher’s algorithm [25] which optimality was accredited in a broad variety of computational applications [1, 26]. Note that integrating semantic similarity assessment in Chawathe’s algorithm [3] denotes a straightforward integration of semantic similarity in [17]’s approach, the latter being a strict generalization of the former. On the other hand, we adopt [17]’s XML data model (Chawathe [5] considering generic hierarchical structured data), which will be explicitly developed in following paragraphs. In fact, we are in agreement with [8, 17]’s decision to disregard element/attribute values while focusing on the structural

---

1 Note that the information content of a concept/class is approximated by estimating the probability of occurrence of the concept/class words in a text corpus.
2 Following Lin [13], the commonality between two concepts is underlined by the information content of their lowest common ancestor (identified by Resnick’s measure [20]). However, the difference between two concepts depends on their own information contents (which are overlooked by Resnick’s measure [20]). Lin’s measure [13] is developed subsequently.
3 Stems designate the morphological variants of a term: an acronym and its expansions, a singular term and its plural, …
properties of XML documents adding that, in order to compare element/attribute values, corresponding types should be previously known, which requires prior knowledge of related XML schemas (recall that this study focuses on comparing XML documents lacking DTDs/XML Schemas).

3.1.1 Basic definitions

Definition 1 - Ordered Labeled Tree (OLT): It is a rooted tree in which nodes are ordered and labeled. In the rest of this paper, the term tree means OLT (cf. Figure 2).

Definition 2 – First level Sub-tree: Given an ordered tree \( T \) with a root node \( r \) of degree \( k \), the first-level sub-trees, \( T_1, T_2, \ldots, T_k \) of \( T \) are the sub-trees rooted at \( r_1, r_2, \ldots, r_k \).

Chawathe [5] models changes to trees using three basic tree edit operations:

Definition 3 - Insertion: Given a node \( x \) of degree \( 0 \) (leaf node) and a node \( p \) in tree \( T \) with first level sub-trees \( p_1, \ldots, p_m \), \( \text{Ins}(x, i, p, \lambda(x)) \) is a node insertion operation applied to \( p \) at position \( i \) that yields \( p' \) with first level sub-trees \( p_1, \ldots, p_i, x, p_{i+1}, \ldots, p_m \) bearing \( \lambda(x) \) as its label.

Definition 4 - Deletion: Given a leaf node \( x \), \( x \) being the \( i \)th child of \( p \), \( \text{Del}(x, i) \) is a node deletion operation applied to node \( p \) that yields \( p' \) with first level sub-trees \( p_1, \ldots, p_i, \ldots, p_m \).

Definition 5 - Update: Given a node \( x \) in tree \( T \), and given a label \( l \), \( \text{Upd}(x, l) \) is a node update operation applied to \( x \) resulting in \( T' \) which is identical to \( T \) except that in \( T' \), \( \lambda(x) = l \). The update operation could be also formulated as follows: \( \text{Upd}(x, y) \) where \( y.l \) denotes the new label to be assumed by \( \lambda(x) \).

Following [5], we presume that the root of a tree cannot be deleted or inserted.

Definition 6 - Edit Script: An edit script \( ES \) is a sequence of edit operations. When applied to a tree \( T \), the resulting tree \( T' \) is obtained by applying edit operations of \( ES \) to \( T \), following their order of appearance in the script.

By associating costs with each edit operation, Chawathe [5] defines the cost of an edit script to be the sum of the costs of its component operations. The author in [5] subsequently states the problem of comparing trees: Given two rooted, labeled, ordered trees \( A \) and \( B \), find a minimum cost edit script that transforms \( A \) to a tree that is isomorphic to \( B \). Note that two trees are said to be isomorphic if they are identical except for node identifiers.

3.1.2 Structural similarity algorithm

In [5], Chawathe employed edit graphs in his edit distance process. However, our study of the edit distance algorithm literature showed that the edit graph used in [5] is a direct application of the famous Wagner-Fisher algorithm [25], updated to take into account tree structures (the Wagner-Fisher algorithm being originally designed for sequence/string comparisons). Therefore, we propose to develop Chawathe’s algorithm [5], using the Wagner-Fisher algorithm [25], and introducing Chawathe’s tree structure updates.

Before proceeding, let us report the ld-pair representation of a tree node introduced in [5]. It is defined as the pair \((l, d)\) where: \( l \) and \( d \) are respectively the node’s label and depth in the tree. As in [5], we use \( p.l \) and \( p.d \) to refer to the label and the depth of an ld-pair \( p \) respectively. Subsequently, the ld-pair representation of a tree is the list, in preorder, of the ld-pairs of its nodes (cf. Figure 3). In [5]’s process, trees are always treated in their ld-pair representations. Given a tree in ld-pair representation \( A = (a_1, a_2, \ldots, a_n) \), \( A[i] \) refers to the \( i \)th node of tree \( A \). Consequently, \( A[i].l \) and \( A[i].d \) denote, respectively, the label and the depth of the \( i \)th node of \( A \).

\[
\begin{align*}
A &= ((\text{Academy}, 0), (\text{Department}, 1), (\text{Laboratory}, 2), (\text{Professor}, 3), (\text{Student}, 3)) \\
B &= ((\text{College}, 0), (\text{Student}, 3), (\text{Professor}, 3)) \\
C &= ((\text{Factory}, 0), (\text{Laboratory}, 2), (\text{Student}, 3))
\end{align*}
\]

Fig. 3. Ld-pair representation of XML sample trees \( A, B \) and \( C \)

The edit distance algorithm, employed in this study, is developed in Figure 4. The ld-pair representation as well
as the added conditions make up Chawathe’s updates [5] to the classic edit distance approach [25]. Chawathe [5] succeeded in transforming trees into modified sequences (ld-pairs), making them suitable for standard edit distance computations. He subsequently added specific conditions so that the edit distance process could take into account tree structures:

- **Condition 1**: limits update operations to nodes having identical depths
- **Condition 2**: intuitively implies that, in order to delete an internal node, all corresponding descendant nodes must be first deleted
- **Condition 3**: implies that, a node must be inserted before inserting any of its descendents

![Fig. 4. Structural similarity algorithm](image)

Note that the distance value between two trees A and B denotes, in a roundabout way, the similarity between them (the smaller the distance between A and B, the more similar they are).

\[
\text{Sim}(A, B) = \frac{1}{1 + \text{Dist}(A, B)} \quad (1)
\]

Similarity measures based on edit (or metric) distance are generally computed as in (1), conforming to the formal definition of similarity [7]:

- \( \text{Sim}(x, y) = 0 \Rightarrow x = y \) \( \text{and} \ y \) are identical\(^1\).
- \( \text{Sim}(x, y) = 0 \Rightarrow x \) and \( y \) are different and have no common characteristics.
- \( \text{Sim}(x, y) = 1 \Rightarrow \) similarity is reflexive.
- Similarity and distance are inverse to each other.
- \( \text{Sim}(x, y) = \text{Sim}(y, x) \Rightarrow \) similarity is symmetric (Note that symmetry is controversially discussed [7] and is domain and application-oriented\(^3\)).
- \( \text{Sim}(x, z) \leq (\text{Sim}(x, y) + \text{Sim}(y, z)) \Rightarrow \) Triangular inequality (as with symmetry, triangular inequality is not always true\(^3\)).

On the other hand, a central question in most edit distance approaches is how to choose operation cost values. An intuitive and natural way would be to assign identical costs to insertion and deletion operations (\(\text{Cost}_{\text{ins}} = \text{Cost}_{\text{del}} = 1\)), as well as to update operations only when the newly assigned label is different from the node’s current label (\(\text{Cost}_{\text{upd}}(a, b) = 1\) when \(a \neq b\)), otherwise, when the labels are the same, \(\text{Cost}_{\text{upd}} = 0\), underlining that no changes are to be made to the label of node \(a\). By applying the preceding intuitive cost model (ICM), the edit distance between XML sample trees \(A\) and \(B\), \(\text{Dist}(A, B)\), would be equal to 3. It is the cost of the following edit script:


The corresponding edit distance computations are shown in Table 1. The minimum-cost ES contribution to the edit distance computation process is emphasized in bold format. Note that an identical edit distance result is attained when comparing sample documents \(A\) and \(C\) (\(\text{Dist}(A, C) = 3\)).

**Tab. 1.** Computing edit distance for XML trees \(A\) and \(B\)\(^4\)

<table>
<thead>
<tr>
<th></th>
<th>Dist(A, B)</th>
<th>Dist(A, C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>A[1]</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>A[2]</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>A[4]</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>A[5]</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

As previously mentioned in our **motivation** paragraph, comparing sample documents \(A\), \(B\) and \(C\), via strict

---

1 This property isn’t always verified in the literature [14]. It depends on the chosen similarity measure. However, \(x = y \Rightarrow \text{Sim}(x, y) = 1\) is true regardless of the measure employed.
2 Several authors have proposed asymmetric measures [9, 14].
3 Both symmetry and triangular inequality will be discussed in Section 4.
4 In the edit distance computational tables developed throughout the paper, node labels are abbreviated (i.e. prof instead of professor) due to paper format constraints.
structural evaluation, yields identical similarity values, the semantics involved being disregarded:

\[ \text{Sim}(A, B) = \text{Sim}(A, C) = 1/(1+3) = 0.25 \]

In order to amend precision and accuracy of XML similarity, we propose the use of an original cost scheme, integrating IR semantic relatedness in the structure-based similarity algorithm.

### 3.2 Integrated semantic & structure based similarity

Apparently, intuitive cost schemes (like the one used previously) do not affect the correctness of the structural similarity algorithm. However, they fail to capture the semantics of XML documents. In this study, we propose to complement Chawathe’s edit distance approach [5], with a cost scheme integrating semantic assessment.

#### 3.2.1 Semantic similarity measure

Our investigation of the IR semantic similarity literature led us to consider Lin’s similarity measure [13], in our XML comparison process. Lin’s measure was proven efficient in evaluating semantic similarity. Its performance and theoretical basis are recognized and generalized by [15] to deal with hierarchical and non-hierarchical structures. Please bear in mind that our XML generalized by [15] to deal with hierarchical and non-hierarchical structures. Please bear in mind that our XML comparison process is not sensitive, in its definition, to the hierarchical structures. Therefore, overall similarity $\text{Sim}(A, B)$ should be of greater value vis-à-vis $\text{Sim}(A, C)$. Such semantic relatedness would be taken into consideration by varying operation costs as follows:

$$\text{Cost}_{\text{Sem-Upd}}(x, y) = 1 - \text{Sim}_{\text{Sem}}(x.l, y.l)$$  \hfill (3)

The more the initial and the replacing node labels ($x.l$ and $y.l$ respectively) are semantically similar, the lesser the update operation cost, which transitively yields a lesser minimum cost $ES$ (higher similarity value). When labels are identical, semantic similarity is of maximum value, $\text{Sim}_{\text{Sem}}(x.l, y.l) = 1$, yielding $\text{Cost}_{\text{Upd}}(x, y) = 0$ (no changes to be made). When labels are completely different, semantic similarity is of minimum value, $\text{Sim}_{\text{Sem}}(x.l, y.l) = 0$, which brings us to $\text{Cost}_{\text{Upd}}(x, y) = 1$. Following the same logic, we consider varying insertion and deletion costs.

$$\text{Cost}_{\text{Sem-Ins}}(x, i, p, \lambda(x)) = 1 - \text{Sim}_{\text{Sem}}(\lambda(x), p.l)$$  \hfill (4)

$$\text{Cost}_{\text{Sem-De}}(x, p) = 1 - \text{Sim}_{\text{Sem}}(x.l, p.l)$$  \hfill (5)

While inserting or deleting a node from an XML document, we evaluate semantic relatedness between the inserted/deleted node’s label and the label of its ancestor in the document tree. The more an inserted/deleted node label is semantically similar to its ancestor node label, the lesser the insertion/deletion operation cost, which transitively yields a lesser cost $ES$ (higher similarity value). When labels are identical or completely different, insertion/deletion costs would be equal to 0 or 1,
respectively\(^1\) (as with the update operation). Such semantic assessments would reflect semantic relatedness between inserted/deleted nodes and their context, in the XML document, affecting overall similarity accordingly. Furthermore, our investigations of semantic similarity, in XML documents, led us to consider varying operation costs with respect to node depth.

### 3.2.3 Node depth cost

Node depth consideration in XML document comparison is not original in the literature. Zhang et al. [29] have already addressed the issue. Following [29], editing the root node of an XML tree would yield significantly greater change than editing a leaf node. Notationally, as one descends in the XML tree hierarchy, information becomes increasingly specific, consisting of finer and finer details, its affect on the whole document tree decreasing accordingly. For example, consider the XML sample tree \(A\) in Figure 2. Editing node \(A[1]\) (\(A[1].l = \text{Academy}\)) by changing its label to \(\text{Hospital}\), would semantically affect tree \(A\) a lot more than deleting node \(A[4]\) (\(A[4].l = \text{Professor}\)), changing \(A\)'s whole semantic context. Therefore, it would be relevant to vary operation costs following node depths, assuming that operations near the root node have higher impact than operations further down the hierarchy. The following formula, adapted from [29], could be used for that matter:

\[
\text{Cost}_{\text{Depth, } \text{Op}}(x) = \frac{1}{(1 + x.d)} 
\]

\(^6\)

- \(\text{Op}\) is an insert, delete or update operation
- \(x.d\) is the depth of the node considered for insertion, deletion or updating

The preceding formula assigns unit cost (=1, maximum cost) when the root node is considered and yields decreasing costs when moving downward in the hierarchy.

### 3.2.4 Semantic cost model

In order to take into account semantic meaning while comparing XML documents, we propose to complement Chawathe’s edit distance algorithm [5], with the following cost model:

\[
\text{Cost}_{\text{Op}}(x, y) = \text{Cost}_{\text{Sem, Op}}(x, y) \times \text{Cost}_{\text{Depth, Op}}(x) 
\]

\(^7\)

- \(\text{Op}\) denotes an insertion, deletion or update operation

The results attained by applying the semantic cost model to compare sample XML documents \(A\), \(B\) and \(C\) are shown in tables 2 and 3. Note that semantic similarity values between node labels were estimated using Lin’s measure [13] (applied on an independently constructed corpus and taxonomy), and are reported in Table 4.

### Tab. 2. Computing edit distance, via our SCM, for XML sample trees \(A\) and \(B\)

<table>
<thead>
<tr>
<th>(0)</th>
<th>(0)</th>
<th>(1)</th>
<th>(1.5)</th>
<th>(1.8333)</th>
<th>(2.0833)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A[1]) (Med, 0)</td>
<td>1</td>
<td>(0.1148)</td>
<td>0.5365</td>
<td>0.8205</td>
<td>0.9824</td>
</tr>
<tr>
<td>(A[2]) (Dept, 0)</td>
<td>1.2171</td>
<td>0.5365</td>
<td>(0.1148)</td>
<td>0.1423</td>
<td>0.3413</td>
</tr>
<tr>
<td>(A[3]) (Lab, 0)</td>
<td>1.4994</td>
<td>0.5642</td>
<td>0.1423</td>
<td>(0.1148)</td>
<td>0.3172</td>
</tr>
<tr>
<td>(A[4]) (Prof, 0)</td>
<td>1.651</td>
<td>1.7638</td>
<td>0.3441</td>
<td>0.3164</td>
<td>(0.163)</td>
</tr>
<tr>
<td>(A[5]) (Std, 0)</td>
<td>1.8466</td>
<td>1.9614</td>
<td>0.5397</td>
<td>0.512</td>
<td>(0.3586)</td>
</tr>
</tbody>
</table>

### Tab. 3. Computing edit distance, via our SCM, for XML trees \(A\) and \(C\)

<table>
<thead>
<tr>
<th>(0)</th>
<th>(0)</th>
<th>(1)</th>
<th>(1.5)</th>
<th>(1.8333)</th>
<th>(2.0833)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A[1]) (Med, 0)</td>
<td>1</td>
<td>(0.8581)</td>
<td>1.2798</td>
<td>1.5638</td>
<td>1.7813</td>
</tr>
<tr>
<td>(A[2]) (Dept, 0)</td>
<td>1.2171</td>
<td>1.2798</td>
<td>(0.8581)</td>
<td>0.8581</td>
<td>1.0989</td>
</tr>
<tr>
<td>(A[3]) (Lab, 0)</td>
<td>1.4994</td>
<td>1.3075</td>
<td>0.8581</td>
<td>(0.8581)</td>
<td>1.0647</td>
</tr>
<tr>
<td>(A[4]) (Prof, 0)</td>
<td>1.651</td>
<td>1.5921</td>
<td>1.0874</td>
<td>1.0597</td>
<td>(1.0673)</td>
</tr>
<tr>
<td>(A[5]) (Std, 0)</td>
<td>1.8466</td>
<td>1.7047</td>
<td>1.283</td>
<td>1.2553</td>
<td>(1.2628)</td>
</tr>
</tbody>
</table>

By applying our SCM, the edit distances computed between pairs \(A-B\) and \(A-C\) are no longer identical (in comparison with the intuitive cost scheme):

\[
\text{Sim}_{\text{SCM}}(A, B) = \frac{1}{(1 + \text{Dist}_{\text{SCM}}(A, B))} = 0.7361 \text{ having}\]
\[
\text{Dist}_{\text{SCM}}(A, B) = 0.3586
\]

\[
\text{Sim}_{\text{SCM}}(A, C) = \frac{1}{(1 + \text{Dist}_{\text{SCM}}(A, C))} = 0.4418 \text{ having}\]
\[
\text{Dist}_{\text{SCM}}(A, C) = 1.2628
\]

Considering semantic relatedness, in the comparison process, reflects the fact that sample documents \(A\) and \(B\) are more similar than \(A\) and \(C\) (\(\text{Sim}_{\text{SCM}}(A, B) > \text{Sim}_{\text{SCM}}(A, C)\)), in spite of sharing identical structural similarities.

Our SCM, used with a structure-based (edit distance) similarity algorithm, seems to capture semantic meaning effectively, while comparing XML documents.

### 4. Discussion

Similarity is a fundamental concept in many fields, e.g. information retrieval, and is commonly used in multidimensional data processing and viewed as a relation satisfying certain properties [15]. The formal definition of similarity, given in [7] (cf. Section 3.1.2), identifies such

\(^1\) In this study, we assume that an XML node and its ancestor cannot have identical labels. However, such cases this will be addressed in future work.
properties which can be viewed as a concrete explanation of the generally abstract concept of similarity.

**Tab. 4.** Word semantic similarities, computed following Lin’s measure [11]

<table>
<thead>
<tr>
<th>Word pairs</th>
<th>SimLin</th>
<th>Word pairs</th>
<th>SimLin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Academy</td>
<td>0.8851</td>
<td>Department</td>
<td>0.2083</td>
</tr>
<tr>
<td>Academy</td>
<td>0.1566</td>
<td>Department</td>
<td>0.2367</td>
</tr>
<tr>
<td>Academy</td>
<td>0.8419</td>
<td>Department</td>
<td>0.1857</td>
</tr>
<tr>
<td>Academy</td>
<td>0.1481</td>
<td>Factory</td>
<td>0.1965</td>
</tr>
<tr>
<td>Academy</td>
<td>0.3521</td>
<td>Factory</td>
<td>0.1803</td>
</tr>
<tr>
<td>Academy</td>
<td>0.3565</td>
<td>Factory</td>
<td>0.1831</td>
</tr>
<tr>
<td>Academy</td>
<td>0.3876</td>
<td>Factory</td>
<td>0.2047</td>
</tr>
<tr>
<td>Academy</td>
<td>0.1297</td>
<td>Supervisor</td>
<td>0.4672</td>
</tr>
<tr>
<td>College</td>
<td>0.1566</td>
<td>Laboratory</td>
<td>0.1903</td>
</tr>
<tr>
<td>College</td>
<td>0.8419</td>
<td>Laboratory</td>
<td>0.1935</td>
</tr>
<tr>
<td>College</td>
<td>0.1481</td>
<td>Laboratory</td>
<td>0.2177</td>
</tr>
<tr>
<td>College</td>
<td>0.3521</td>
<td>Laboratory</td>
<td>0.1738</td>
</tr>
<tr>
<td>College</td>
<td>0.3565</td>
<td>Professor</td>
<td>0.847</td>
</tr>
<tr>
<td>College</td>
<td>0.3876</td>
<td>Professor</td>
<td>0.5028</td>
</tr>
<tr>
<td>College</td>
<td>0.1297</td>
<td>Supervisor</td>
<td>0.1611</td>
</tr>
<tr>
<td>Department</td>
<td>0.2117</td>
<td>Professor</td>
<td>0.5114</td>
</tr>
<tr>
<td>Department</td>
<td>0.9149</td>
<td>Supervisor</td>
<td>0.1633</td>
</tr>
<tr>
<td>Department</td>
<td>0.2047</td>
<td>Student</td>
<td>0.1803</td>
</tr>
</tbody>
</table>

Therefore, a newly introduced similarity method, such as the one developed in this paper, should be normally evaluated w.r.t to the formal definition of similarity [7] in order to assess its consistency with the similarity concept. Our combined semantic and structure based XML similarity approach follows the formal definition of similarity [7] except for symmetricity and triangular inequality which are debated in IR research [13, 14, 15]. Those two properties will be detailed below, the remaining similarity properties being obvious (cf. Section 3.1.2).

### 4.1 Symmetricity

Despite combining symmetric edit distance [5] and semantic similarity [13] measures, our approach is asymmetric, that is SimSCM(A, B) ≠ SimSCM(B, A). Consider for example XML trees D and F in Figure 5.

![XML ordered labeled trees](image)

**Fig. 5.** XML ordered labeled trees

Edit distance computations, using Section 3.1.2’s intuitive cost model (Chawathe’s classical approach [5]), yield the following values:

- SimICM(D, F) = SimICM(D, F) = 0.25 having DistICM(D, F) = DistICM(F, D) = 3
- Edit script(D, F) : Upd(D[1], F[1]), Del(D[2], D[1]), Del(D[3], D[1])
- Edit script(F, D) : Upd(F[1], D[1]), Ins(D[2], 1, F[1], Professor), Ins(D[3], 2, F[1], Student)

On the other hand, when using our SCM, similarity values become as follows:

- SimSCM(D, F) = 0.4022 > SimSCM(F, D) = 0.3753

That is due to the varying semantic costs of insert/delete operations. In traditional cost models (e.g. the ICM considered in this paper), insert/delete operations are treated equally (costDel = costIns). However, insert/delete operation costs, in our SCM, depend on the semantic relatedness between the node label being inserted/deleted and the label of its ancestor in the document tree. Therefore:

- Costsem,Del(D[2], D[1]) = 1 − SimSem(Professor, Academy) = 0.6437
- Costsem,Ins(D[2], 1, F[1], Professor) = 1 − SimSem(Professor, Factory) = 0.8169

Likewise for remaining insert/delete operations, which yield different overall ES costs (hence similarity values) for D/F and F/D transformations respectively. In other words, deleting nodes D[2] (Professor) and D[3] (Student) form ancestor D[1] (Academy)’s sub-tree does not affect, semantically, tree D as much as inserting those nodes in tree F, under F[1] (Factory). That is because labels Professor and Student are relatively more similar to label Academy than to Factory. Therefore, D[2] and D[3]’s deletions do not induce a major change in tree D’s meaning. However, their insertions under root node F[1] (Factory) introduce relatively new semantic meaning to tree F, since their labels are relatively dissimilar to Factory (cf. Table 4).

Nevertheless, as mentioned earlier in Section 3.1.2, we keep in mind that symmetricity is widely discussed [7] and might prove to be useful, depending on the nature of the XML-based data being compared, as well as the scenario at hand. Therefore, in cases where asymmetricity is inadequate, a symmetric score, between XML trees D and F for example, can be defined as the arithmetic mean of the two asymmetric scores (as with the average similarity degree measure utilized in our experimental evaluation, cf. Section 5.2).

\[
\text{Ave}(D, F) = \frac{\text{Sim}(D, F) + \text{Sim}(F, D)}{2} \tag{8}
\]

### 4.2 Triangular inequality

While triangular inequality is an axiom for metric distance functions, and is verified for our edit distance approach (SimSem(A, C) ≤ SimSem(A, B) + SimSem(B, C) considering sample XML documents A, B and C), and despite appearing to be intuitive, it is not always true.
Lin’s similarity measure, as well as most semantic similarity measures proposed in the literature [13, 15, 20], do not satisfy triangular inequality:

\[ \text{Sim}_{\text{sc}}(x, z) \leq (\text{Sim}_{\text{sc}}(x, y) + \text{Sim}_{\text{sc}}(y, z)) \]  

(9)

Triangular inequality does not seem to be proper for semantic similarity measures. An example by Tversky [24], reported by Maguitman [15] illustrates the impropriety of triangular inequality with an example about the similarity between countries: “Jamaica is similar to Cuba (because of geographical proximity); Cuba is similar to Russia (because of their political affinity); but Jamaica and Russia are not similar at all”. Since we take into account semantic similarity (between XML element/attribute tags) via Lin’s measure [13], in our semantic cost model SCM, our integrated semantic/structural approach does not transitively satisfy triangular inequality (in agreement with existing semantic similarity approaches [13, 15, 20]).

5. Experimental evaluation

5.1 Prototype

To validate our approach, we have implemented (using C#) a prototype, entitled “XML SS Similarity” (XS5), encompassing a validation component, verifying the integrity of XML documents, and an edit distance component undertaking XML similarity computations following the algorithm adopted in our study. In addition, a synthetic XML data generator was also implemented in order to produce sets of XML documents based on given DTDs. The synthetic XML generator accepts as input: a DTD document and a MaxRepeats1 value designating the maximum number of times a node will appear as child of its parent (when * or + options are encountered in the DTD). Furthermore, a taxonomic analyzer was also introduced so as to compute semantic similarity values between words (expressions) in a given taxonomy. Our taxonomic analyzer accepts as input a hierarchical taxonomy and corresponding corpus-based word occurrences. Consequently, concept frequencies are computed and, thereafter, used to compute semantic similarity between pairs of nodes in the knowledge base.

5.2 Experimental results

Various experiments were conducted in order to test the performance of our integrated similarity model. Real and generated (synthetic) XML documents as well as a number of hierarchical taxonomies where considered. In the following, we present the results attained using synthetic XML documents (cf. Figure 6) and a WordNet2 based hierarchical taxonomy comprising of 677 nodes.

Fig. 6. DTDs inducing sets of synthesized XML documents

We evaluate our model’s efficiency by assessing similarity results to the a priori know DTDs (inducing document sets). Therefore, average inter-set and intra-set similarities are depicted in a matrix where element \((i, j)\) underscores the average similarity value, \(\text{Sim}(S_i, S_j)\), corresponding to every pair of distinct documents such that the first belongs to the set \(S_i\) (DTDi) and the second to the set \(S_j\) (DTDj).

Note that the asymmetry of our approach is reflected by the intra-set similarity values: \(\text{Sim}(S_i, S_j) \neq \text{Sim}(S_j, S_i)\) using our SCM, while symmetry is preserved using the ICM (Chawathe’s classical approach [5]) (cf. tables 5 and 6).

---

1 A greater MaxRepeats value underlines a greater variability when + and * are encountered.

2 WordNet is an online lexical reference system (taxonomy), developed by a group of researchers at Princeton University NJ USA, where nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing a lexical concept [16].

3 Intra-set similarities are computed between documents of the same set \(S_r\), reported as \((i, i)\) values in the similarity matrix. Remaining \((i, j)\) values correspond to intra-set similarities, computed between documents belonging to sets \(S_i\) and \(S_j\)
First of all, results show that our SCM produces higher similarity values, in comparison with the ICM, underlining similarities (of semantic nature) that were undetected using the latter. On the other hand, a straight distinction between documents belonging to a set and others outside that set is attained with our SCM, as with the ICM (comparing highlighted values, in tables 5 and 6, remaining values).

Furthermore, our SCM captures semantic affinities between documents corresponding to different sets, inducing changes in the relative ranking between values belonging to the ICM matrix and those corresponding to the SCM matrixes. In order to reflect semantic affinities between XML documents of different sets, we define the average similarity degree between two sets of documents: Ave(S1, S2) as the arithmetic mean of the average intra-set similarity values Sim(S1, S2) and Sim(S2, S1) corresponding to those sets, as given in (8) (thus attaining a symmetric measure for comparing XML document sets). Consequently, we identified a higher average similarity degree between sets S1 and S2 (AveSCM(S1, S2) = 0.3403, DTDs 1 and 2 revealing semantic similarities), using our SCM, in comparison with S1 and S3 (AveSCM(S1, S3) = 0.3325), the average similarity degree between S1/S2 (AveICM(S1, S2) = 0.0951) being less than that of S1/S3 (AveICM(S1, S3) = 0.0982) using the ICM (cf. Table 7, Figure 7).

**Tab. 5. Inter/intra set similarities via ICM**

<table>
<thead>
<tr>
<th></th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>0.5886</td>
<td>0.0951</td>
<td>0.0982</td>
<td>0.0774</td>
<td>0.0237</td>
</tr>
<tr>
<td>S2</td>
<td>0.0951</td>
<td>0.1515</td>
<td>0.0945</td>
<td>0.0735</td>
<td>0.0234</td>
</tr>
<tr>
<td>S3</td>
<td>0.0982</td>
<td>0.0945</td>
<td>0.4110</td>
<td>0.0732</td>
<td>0.0234</td>
</tr>
<tr>
<td>S4</td>
<td>0.0774</td>
<td>0.0732</td>
<td>0.4164</td>
<td>0.0732</td>
<td>0.0252</td>
</tr>
<tr>
<td>S5</td>
<td>0.0237</td>
<td>0.0234</td>
<td>0.0234</td>
<td>0.0252</td>
<td>0.0981</td>
</tr>
</tbody>
</table>

**Tab. 6. Inter/intra set similarities via SCM**

<table>
<thead>
<tr>
<th></th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>0.8877</td>
<td>0.3407</td>
<td>0.3240</td>
<td>0.2331</td>
<td>0.1104</td>
</tr>
<tr>
<td>S2</td>
<td>0.3400</td>
<td>0.4392</td>
<td>0.3303</td>
<td>0.2238</td>
<td>0.1092</td>
</tr>
<tr>
<td>S3</td>
<td>0.3410</td>
<td>0.3423</td>
<td>0.6400</td>
<td>0.2193</td>
<td>0.1035</td>
</tr>
<tr>
<td>S4</td>
<td>0.1953</td>
<td>0.1905</td>
<td>0.2237</td>
<td>0.7701</td>
<td>0.0987</td>
</tr>
<tr>
<td>S5</td>
<td>0.1674</td>
<td>0.1647</td>
<td>0.2046</td>
<td>0.1644</td>
<td>0.3704</td>
</tr>
</tbody>
</table>

**Tab. 7. Average similarity degrees between S1/S2 & S1/S3**

<table>
<thead>
<tr>
<th></th>
<th>ICM</th>
<th>SCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ave(S1, S2)</td>
<td>0.0951</td>
<td>0.3403</td>
</tr>
<tr>
<td>Ave(S1, S3)</td>
<td>0.0982</td>
<td>0.3325</td>
</tr>
</tbody>
</table>

5.3 Timing analysis

The combined structural/semantic XML similarity results, reached using our SCM, aren’t attained without affecting overall time complexity.

First of all, recall that Chawathe’s edit distance process [3], which we developed in this paper, is linear in the number of nodes of each tree, and polynomial (quadratic) in the size of the two trees being compared: O(|A||B|) (which can be simplified to O(N²), N being the maximum number of nodes in trees A and B). This linear dependency on the size of each tree is experimentally verified, timing results being presented in figures 8 and 9. The timing experiments were carried out on a Pentium 4 PC (2.8 GHz CPU, 798 MHz bus, 512 MB RAM).

One can see that the time to compute similarity grows in an almost perfect linear fashion, when using the classic ICM (cf. Figure 8). However, when introducing our SCM, it incrementally shifts towards a polynomial (quadratic) function, following the growing number of taxonomic nodes involved (cf. Figure 9). Naturally, Figure 9 reflects, not only the time complexity of the edit distance process, but also that of the taxonomic analysis process (SCM).
To our knowledge, time complexity for Lin’s measure [13] was not conducted previously. Therefore, we estimated its complexity via our implementation components: $\text{Depth}(T)^2$ where $T$ is the taxonomy considered and $\text{Depth}(T)$ is the maximum taxonomic depth. Consequently, in order to reduce our model’s overall complexity, we computed semantic similarity for each pair of nodes in the taxonomy considered (which took more than 7 CPU hours), stored semantic similarity results in a dedicated indexed table (Oracle 9i DB)\(^1\), and accessed that table to acquire semantic values when using our SCM (instead of traversing the taxonomy to compute semantic similarity each time it is needed). An average of 0.25 seconds per pair-wise semantic similarity assessment was saved, when exploiting the 677 words WordNet-based taxonomy, owing to that procedure (cf. Figure 9).

6. Conclusion and future work

In this paper, we proposed an integrated semantic and structure based XML similarity approach, taking into account the semantic meaning of XML element/attribute labels in XML document comparison. To our knowledge, this is the first attempt to combine edit distance structural similarity computations with IR semantic similarity assessment, in an XML (structured data) context. Experimental results confirmed the positive impact of semantic meaning on XML similarity values, and reflected its heavy impact regarding complexity.

Future directions include exploiting semantic similarity to compare, not only the structure of XML documents (element/attribute labels), but also their information content (element/attribute values). In such a framework, XML Schemas seem unsurpassable, underlining element/attribute data types, required to compare corresponding element/attribute values. Our future goals will also incorporate studying applied multimedia similarity computations (MPEG7, SVG documents, …), taking into consideration structural, semantic, as well as multimedia-specific criterion (if necessary) while comparing XML-based multimedia documents. The semantic complexity problem will also be tackled in upcoming studies.

References


\(^1\) Oracle uses the B-Tree indexing technique.

Fig. 9. Timing results after introducing our SCM.


