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 symmetricity 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.

Submitted to COMAD 2006.

Proceedings of the 13th International Conference on Management of Data (COMAD) 2006
Delhi, Dec 2006

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*.

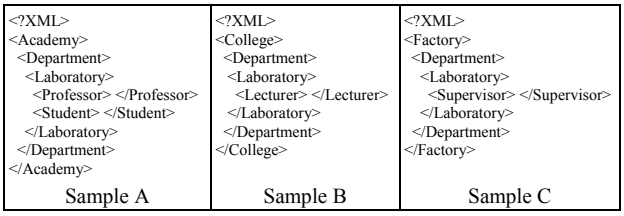

Fig. 1. Examples of XML documents

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| depth(A) 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 $O(N^2)$. 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 in 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 $O(2N+N^2)$.

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[1].

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*[2]. 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[3]) 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 Nierman 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 tow 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*).

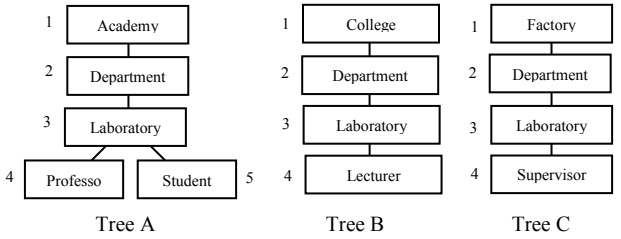

Fig. 2. OLTs corresponding to sample documents *A*, *B* and *C*

The number next to a node is its preorder rank and serves as node identifier. Please note that there is no correspondence between node identifiers when given two trees to compare. Node correspondence can only be achieved through node labels, taking into account their positions in the trees.

Definition 2 – First level Sub-tree: Given an ordered tree *T*, with a root node *r* of degree[1] *k*, the first-level sub-trees, $T_1, T_2, \ldots, T_k$ of *T* are the sub-trees rooted at $r_1, r_2, \ldots, r_k$ [17].

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 $p_1, \ldots, p_m$, *Ins(x, i, p, λ(x))* is a node insertion operation applied to *p* at position *i* that yields *p'* with first level sub-trees $p_1, \ldots, p_i, x, p_{i+1}, \ldots, p_{m+1}$, *x* bearing *λ(x)* as its label.

Definition 4 - Deletion: Given a leaf node *x, x* being the *i*th child of *p*, *Del(x, p)* is a node deletion operation applied to node *p* that yields *p'* with first level sub-trees $p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_m$

Definition 5 - Update: Given a node *x* in tree *T,* and given a label *l*, *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: *Upd(x, y)* where *y.l* denotes the new label to be assumed by *λ(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 a_i 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.

A = ((Academy, 0), (Department, 1), (Laboratory, 2), (Professor, 3), (Student, 3))
B = ((College, 0), (Department, 1), (Laboratory, 2), (Lecturer, 3))
C = ((Factory, 0), (Department, 1), (Laboratory, 2), (Supervisor, 3))

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

[1] The degree of a node *n* underscores the number of sub-trees encompassed by *n*.

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 descendent nodes must be first deleted
- $Condition_3$ implies that, a node must be inserted before inserting any of its descendents

```
Input: Trees A and B (in ld-pair representations)
Output: Edit distance between A and B
Begin
                                                                    1
   Dist [][] = new [0...|A|][0…|B|]
   Dist[0][0] = 0
   For (i = 1 ; i ≤ |A| ; i++) { Dist[i][0] = Dist[i-1][0] + Cost_Del(a_i) }
   For (j = 1 ; j ≤ |B| ; j++) { Dist[0][j] = Dist[0][j-1] + Cost_Ins(b_i) }   5
   For (i = 1 ; i ≤ |A| ; i++)
   {
      For (j = 1 ; j ≤ |B| ; j++)
         {
            Dist[i][j] = min{                                           10
               If (Condition_1 true) { Dist[i-1][j-1] + Cost_Upd(a_i, b_j) }
               If (Condition_2 true) { Dist[i-1][j] + Cost_Del(a_i) }
               If (Condition_3 true) { Dist[i][j-1] + Cost_Ins(b_j) }
                        }
         }                                                              15
   }
   Return Dist[|A|][|B|]      // Distance (similarity) between trees A and B
End
The Chawathe conditions:
Condition_1 { (A[i].d = B[j].d) }
Condition_2 { ( (A[i].d ≥ B[j].d) or (j = |B|) ) }
Condition_3 { ( (A[i].d ≤ B[j].d) or (i = |A|) ) }
```

Fig. 4. Structural similarity algorithm

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).

$$Sim(A, B) = \frac{1}{1 + Dist(A, B)} \qquad \textbf{(1)}$$

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

- $Sim(x, y) \in [0, 1]$.
- $Sim(x, y) = 1 \Rightarrow x = y$ (x and y are identical)[1].
- $Sim(x, y) = 0 \Rightarrow$ x and y are different and have no common characteristics.
- $Sim(x, x) = 1 \Rightarrow$ similarity is reflexive.
- Similarity and distance are inverse to each other.
- $Sim(x, y) = Sim(y, x) \Rightarrow$ similarity is symmetric (Note that symmetricity is controversially discussed [7] and is domain and application-oriented[2]).
- $Sim(x, z) \leq (Sim(x, y) + Sim(y, z)) \Rightarrow$ Triangular inequality (as with symmetricity, 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 ($Cost_{Ins} = Cost_{Del} = 1$), as well as to *update* operations only when the newly assigned label is different from the node's current label ($Cost_{Upd}(a, b) = 1$ when $a.l \neq b.l$, otherwise, when the labels are the same, $Cost_{Upd} = 0$, 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*, *Dist(A, B)*, would be equal to *3*. It is the cost of the following edit script:

- Upd(A[1], B[1]), Upd(A[4], B[4]), Del(A[5], A[3])

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* (*Dist(A, C) = 3*).

Tab. 1. Computing edit distance for XML trees *A* and *B*[4]

	0	B[1] (*Coll.*, 0)	B[2] (*Dept.*, 1)	B[3] (*Lab.*, 2)	B[4] (*Lect.*, 3)
0	0	1	2	3	4
A[1] (*Acd.*, 0)	1	**1**	2	3	4
A[2] (*Dept.*, 1)	2	2	**1**	2	3
A[3] (*Lab.*, 2)	3	3	2	**1**	2
A[4] (*Prof.*, 3)	4	4	3	2	**2**
A[5] (*Std.*, 3)	5	5	4	3	**3**

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, $x = y \Rightarrow Sim(x, y) = 1$ is true regardless of the measure employed.

[2] Several authors have proposed asymmetric measures [9, 14].

[3] Both symmetricity 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:

– Sim(A,B) = 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 similarity process is not sensitive, in its definition, to the semantic similarity measure used. However, choosing a performing measure would yield better similarity judgment.

Following Lin [13], the semantic similarity between two words (expressions) can be computed as:

$$\mathrm{Sim}_{\mathrm{Sem}}(w_1, w_2) = \mathrm{Sim}_{\mathrm{Sem}}(c_1, c_2) = \frac{2 \log p(c_0)}{\log p(c_1) + \log p(c_2)} \quad \textbf{(2)}$$

– c_1 and c_2 are concepts, in a knowledge base of hierarchical structure (taxonomy), subsuming words w_1 and w_2 respectively.
– c_0 is the most specific common ancestor of concepts c_1 and c_2.
– *p(c)* denotes the occurrence probability of words corresponding to concept *c*. It can be computed as the relative frequency: *p(c) = freq(c) / N*.
  - $freq(c) = \sum_{w \in words(c)} \mathrm{count}(w)$: sum of the number of occurrences, of words subsumed by *c*, in a corpus.
  - N: total number of words in the corpus.

In information theory, the *information content* of a class or concept c is measured by the negative log likelihood *-log p(c)* [20, 15]. While comparing two concepts c_1 and c_2, Lin's measure takes into account each concept's *information content* (-log $p(c_1)$ + -log $p(c_2)$), as well as the *information content* of their most specific common ancestor (-log $p(c_0)$), in a way to increase with commonality (*information content* of c_0) and decrease with difference (*information content* of c_1 and c_2) [13]. Lin's measure produces values limited to the [0, 1] interval, and conforms to the formal definition of similarity [7] except for triangular inequality (which will be discussed in Section 4).

3.2.2 Label semantic similarity cost

To take into account semantic similarity in XML comparisons, while utilizing the edit distance algorithm, we propose to vary operation costs according to the semantics of concerned nodes. While comparing XML sample documents *A-B* and *A-C* for example, the similarity evaluation process should realize that elements *Academy-College* have higher semantic similarity than *Academy-Factory*. Likewise, *Professor-Lecturer* have higher semantic similarity than *Professor-Supervisor*. Therefore, overall similarity *Sim(A, B)* should be of greater value vis-à-vis *Sim(A, C)*. Such semantic relatedness would be taken into consideration by varying operation costs as follows:

$$\mathrm{Cost}_{\mathrm{Sem\_Upd}}(x, y) = 1 - \mathrm{Sim}_{\mathrm{Sem}}(x.l, y.l) \quad \textbf{(3)}$$

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, $Sim_{Sem}(x.l, y.l) = 1$, yielding $Cost_{Upd}(x, y) = 0$ (no changes to be made). When labels are completely different, semantic similarity is of minimum value, $Sim_{Sem}(x.l, y.l) = 0$, which brings us to $Cost_{Upd}(x, y) = 1$. Following the same logic, we consider varying insertion and deletion costs.

$$\mathrm{Cost}_{\mathrm{Sem\_Ins}}(x, i, p, \lambda(x)) = 1 - \mathrm{Sim}_{\mathrm{Sem}}(\lambda(x), p.l) \quad \textbf{(4)}$$

$$\mathrm{Cost}_{\mathrm{Sem\_Del}}(x, p) = 1 - \mathrm{Sim}_{\mathrm{Sem}}(x.l, p.l) \quad \textbf{(5)}$$

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. Notionally, 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 = Academy*) by changing its label to *Hospital*, would semantically affect tree *A* a lot more than deleting node *A[4]* (*A[4].l = 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:

$$\mathrm{Cost}_{\mathrm{Depth\_Op}}(x) = \frac{1}{(1 + x.d)} \qquad \textbf{(6)}$$

- *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:

$$\mathrm{Cost}_{\mathrm{Op}}(x, y) = \mathrm{Cost}_{\mathrm{Sem\_Op}}(x, y) \times \mathrm{Cost}_{\mathrm{Depth\_Op}}(x) \qquad \textbf{(7)}$$

[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.

- *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*

	0	B[1] (*Coll.*, 0)	B[2] (*Dept.*, 1)	B[3] (*Lab.*, 2)	B[4] (Lect., 3)
0	0	1	1.5	1.8333	2.0833
A[1] (*Acd.*, 0)	1	**0.1148**	0.5365	0.8205	0.9824
A[2] (*Dept*, 1)	1.4217	0.5365	**0.1148**	0.1425	0.3413
A[3] (*Lab.*, 2)	1.4494	0.5642	0.1425	**0.1148**	0.3172
A[4] (*Prof.*, 3)	1.651	1.7658	0.3441	0.3164	**0.163**
A[5] (*Std.*, 3)	1.8466	1.9614	0.5397	0.512	**0.3586**

Tab. 3. Computing edit distance, via our SCM, for XML trees *A* and *C*

	0	B[1] (*Fact.*, 0)	B[2] (*Dept.*, 1)	B[3] (*Lab.*, 2)	B[4] (*Sup.*, 3)
0	0	1	1.5	1.8333	2.0833
A[1] (*Acd.*, 0)	1	**0.8581**	1.2798	1.5638	1.7813
A[2] (*Dept.*, 1)	1.4217	1.2798	**0.8581**	0.8858	1.0894
A[3] (*Lab.*, 2)	1.4494	1.3075	0.8858	**0.8581**	1.0647
A[4] (*Prof.*, 3)	1.651	1.5091	1.0874	1.0597	**1.0673**
A[5] (*Std.*, 3)	1.8466	1.7047	1.283	1.2553	**1.2628**

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):

- $\mathrm{Sim}_{\mathrm{SCM}}(A, B) = \frac{1}{(1 + \mathrm{Dist}_{\mathrm{SCM}}(A, B))} = 0.7361$ having $\mathrm{Dist}_{\mathrm{SCM}}(A, B) = 0.3586$
- $\mathrm{Sim}_{\mathrm{SCM}}(A, C) = \frac{1}{(1 + \mathrm{Dist}_{\mathrm{SCM}}(A, C))} = 0.4418$ having $\mathrm{Dist}_{\mathrm{SCM}}(A, C) = 1.2628$

Considering semantic relatedness, in the comparison process, reflects the fact that sample documents *A* and *B* are more similar than *A* and *C* ($Sim_{SCM}(A, B) > Sim_{SCM}(A,C)$), 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

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]

Word pairs		Sim_{Lin}	Word pairs		Sim_{Lin}
Academy	College	0.8851	Department	Professor	0.2083
Academy	Department	0.1566	Department	Student	0.2367
Academy	Factory	0.1419	Department	Supervisor	0.1857
Academy	Laboratory	0.1481	Factory	Laboratory	0.1963
Academy	Lecturer	0.3521	Factory	Lecturer	0.1803
Academy	Professor	0.3563	Factory	Professor	0.1831
Academy	Student	0.3876	Factory	Student	0.2047
Academy	Supervisor	0.1297	Factory	Supervisor	0.4672
College	Department	0.1566	Laboratory	Lecturer	0.1903
College	Factory	0.1419	Laboratory	Professor	0.1935
College	Laboratory	0.1481	Laboratory	Student	0.2177
College	Lecturer	0.3521	Laboratory	Supervisor	0.1738
College	Professor	0.3563	Lecturer	Professor	0.807
College	Student	0.3876	Lecturer	Student	0.5028
College	Supervisor	0.1297	Lecturer	Supervisor	0.1611
Department	Factory	0.2117	Professor	Student	0.5114
Department	Laboratory	0.9169	Professor	Supervisor	0.1633
Department	Lecturer	0.2047	Student	Supervisor	0.1803

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 $Sim_{SCM}(A,B) \neq Sim_{SCM}(B,A)$. Consider for example XML trees *D* and *F* in *Figure 5*.

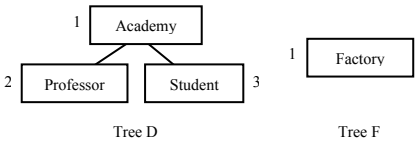

Fig. 5. XML ordered labeled trees

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

- $Sim_{ICM}(D, F) = Sim_{ICM}(D, F) = 0.25$ having $Dist_{ICM}(D, F) = Dist_{ICM}(F, D) = 3$
- 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:

- $Sim_{SCM}(D, F) = 0.4022 > Sim_{SCM}(F, D) = 0.3753$

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 ($cost_{Ins} = cost_{Del}$). 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:

- $Cost_{Sem\_Del}(D[2], D[1]) = 1 - Sim_{Sem}$(Professor, Academy) = 0.6437
- $Cost_{Sem\_Ins}(D[2], 1, F[1],$ Professor$) = 1 - Sim_{Sem}$(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).

$$Ave(D, F) = \frac{(Sim(D, F) + Sim(F, D))}{2} \quad \textbf{(8)}$$

4.2 Triangular inequality

While *triangular inequality* is an axiom for metric distance functions, and is verified for our edit distance approach ($Sim_{Sem}(A, C) \leq Sim_{Sem}(A, B) + Sim_{Sem}(B, C)$ 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*:

$$Sim_{Sem}(x, z) \leq (Sim_{Sem}(x, y) + Sim_{Sem}(y, z)) \quad \textbf{(9)}$$

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" (*XS*[3]), encompassing a v*alidation 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 MaxRepeats[1] 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 WordNet[2] based hierarchical taxonomy comprising of 677 nodes.

```
<!DOCTYPE DTD1 [
<!ELEMENT Academy (Administrative unit+)>
   <!ELEMENT Administrative unit (Branch?)>
      <!ELEMENT Branch (Educator?, Student+)>
         <!ELEMENT Educator (#PCDATA)>
         <!ELEMENT Student (#PCDATA)>  ]>
```

```
<!DOCTYPE DTD2 [
<!ELEMENT School (Administrative unit+)>
   <!ELEMENT Administrative unit (Section?)>
      <!ELEMENT Section (Educator?, Scholar*)>
         <!ELEMENT Educator (#PCDATA)>
         <!ELEMENT Scholar (#PCDATA)>  ]>
```

```
<!DOCTYPE DTD3 [
<!ELEMENT Government (Administrative unit+)>
   <!ELEMENT Administrative unit (Section?)>
      <!ELEMENT Section (Professional?, Worker+)>
         <!ELEMENT Professional (#PCDATA)>
         <!ELEMENT Worker (#PCDATA)>  ]>
```

```
<!DOCTYPE DTD4 [
<!ELEMENT Student (Academic degree*, Educational institution+,
Studies, Experience*, Perspective?)>
   <!ELEMENT Academic degree (#PCDATA)>
   <!ELEMENT Educational institution (#PCDATA)>
   <!ELEMENT Studies (#PCDATA)>
   <!ELEMENT Experience (#PCDATA)>
   <!ELEMENT Perspective (#PCDATA)>  ]>
```

```
<!DOCTYPE DTD5 [
<!ELEMENT Epistemology (Science+)>
   <!ELEMENT Science (Scientists)>
      <!ELEMENT Scientists (Publication?)>
         <!ELEMENT Publication (Document*, Book*, Encyclopedia?)>
            <!ELEMENT Document (#PCDATA)>
            <!ELEMENT Book (#PCDATA)>
            <!ELEMENT Encyclopedia (#PCDATA)>  ]>
```

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[3] similarities are depicted in a matrix where element (i, j) underscores the average similarity value, $Sim(S_i, S_j)$, corresponding to every pair of distinct documents such that the first belongs to the set S_i (DTD_i) and the second to the set S_j (DTD_j).

Note that the asymmetricity of our approach is reflected by the intra-set similarity values: $Sim(S_i, S_j) \neq Sim(S_j, S_i)$ using our SCM, while symmetricity 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_i, 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

Tab. 5. Inter/intra set similarities via ICM

	S1	S2	S3	S4	S5
S1	0.5886	0.0951	0.0982	0.0774	0.0237
S2	0.0951	0.1515	0.0945	0.0735	0.0234
S3	0.0982	0.0945	0.4110	0.0732	0.0234
S4	0.0774	0.0735	0.0732	0.4164	0.0252
S5	0.0237	0.0234	0.0234	0.0252	0.0981

Tab. 6. Inter/intra set similarities via SCM

	S1	S2	S3	S4	S5
S1	0.8877	0.3407	0.3240	0.2331	0.1104
S2	0.3400	0.4392	0.3303	0.2238	0.1092
S3	0.3410	0.3423	0.6400	0.2193	0.1035
S4	0.1953	0.1905	0.2337	0.7701	0.0987
S5	0.1674	0.1647	0.2046	0.1644	0.4704

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 $Sim(S_1, S_2)$ and $Sim(S_2, S_1)$ 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 S_1 and S_2 ($Ave_{SCM}(S1, S2) = 0.3403$, DTDs *1* and *2* revealing semantic similarities), using our SCM, in comparison with S_1 and S_3 ($Ave_{SCM}(S1, S3) = 0.3325$), the average similarity degree between S_1/S_2 ($Ave_{ICM}(S1, S2) = 0.0951$) being lesser than that of S_1/S_3 ($Ave_{ICM}(S1, S3) = 0.0982$) using the ICM (cf. *Table 7, Figure 7*).

Tab. 7. Average similarity degrees between S_1/S_2 & S_1/S_3

	ICM	SCM
Ave(S1, S2)	0.0951	0.3403
Ave(S1, S3)	0.0982	0.3325

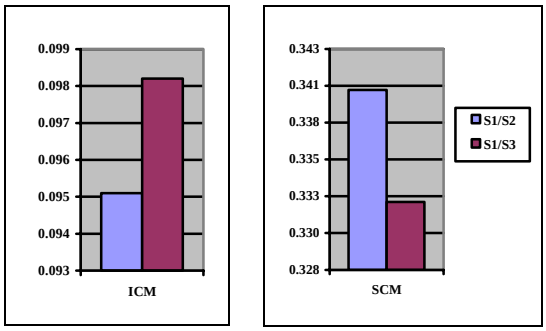

Fig. 7. Average similarity degrees between sets S_1/S_2 and S_1/S_3 – graphical representation.

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 $O(N^2)$, 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).

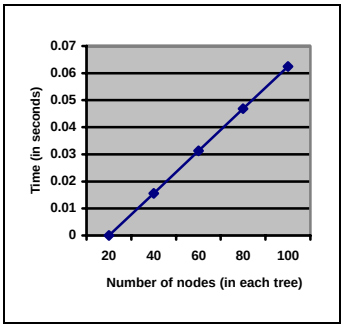

Fig. 8. Timing results while using the ICM

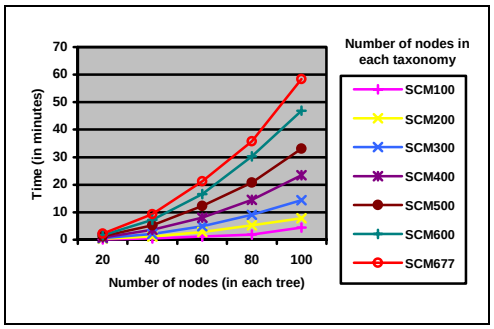

Fig. 9. Timing results after introducing our SCM

To our knowledge, time complexity for Lin's measure [13] was not conducted previously. Therefore, we estimated its complexity via our implementation components: $Depth(T)^2$ where T is the taxonomy considered and 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 futur 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.

[1] Oracle uses the *B-Tree* indexing technique

References

1. Aho A., Hirschberg D., and Ullman J., Bounds on the Complexity of the Longest Common Subsequence Problem. *Journal of the Association for Computing Machinery*, 23(1):1-12, January 1976.
2. Bertino E., Guerrini G., Mesiti M., Rivara I. and Tavella C., Measuring the Structural Similarity among XML Documents and DTDs, Technical Report, University of Genova, 2002, http://www.disi.unige.it/person/MesitiM.
3. Bertino E., Guerrini G., Mesiti M., A Matching Algorithm for Measuring the Structural Similarity between an XML Documents and a DTD and its Applications, *Elsevier Computer Science*, 29 (23-46), 2004.
4. Chawathe S., Rajaraman A., Garcia-Molina H., and Widom J., Change Detection in Hierarchically Structured Information. In *Proceedings of the ACM Int. Conf. on Management of Data (SIGMOD)*, Montreal, Canada, 1996.
5. Chawathe S., Comparing Hierarchical Data in External Memory. In *Proceedings of the Twenty-fth International Conference on Very Large Data Bases,* p. 90-101, 1999.
6. Cobéna G., Abiteboul S. and Marian A., Detecting Changes in XML Documents. In *Proc. of the IEEE Int. Conf. on Data Engineering*, p. 41-52, 2002.
7. EHRIG M. and SURE Y., Ontology Mapping - an Integrated Approach. In *Proceedings of the First European Semantic Web Symposium*, volume 3053 of LNCS, pages 76-91, Heraklion, Greece, 2004
8. Flesca S., Manco G., Masciari E., Pontieri L., and Pugliese A., Detecting Structural Similarities Between XML Documents. In *Proceedings of WebDB 2002*, 2002.
9. Ganesan P., Garcia-Molina H. And Windom J., Exploiting Hierarchical Domain Structure To Compute Similarity. *ACM Transactions on Information Systems (TOIS), Volume 21, Issue 1, 64 – 93, 2003*
10. Jiang J. and Conrath D., Semantic Similarity based on Corpus Statistics and Lexical Taxonomy. *In Proceedings of the International Conference on Research in Computational Linguistics*, 1997.
11. Lee J.H., Kim M.H. and Lee Y.J., Information Retrieval Based on Conceptual Distance in IS-A Hierarchies. *Journal of Documentation*, 49(2):188-207, 1993.
12. Levenshtein V., Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Sov. Phys. Dokl.*, 6:707-710, 1966.
13. Lin D., An Information-Theoretic Definition of Similarity. *In Proceedings of the 15th International Conference on Machine Learning*, 296-304, 1998.
14. Ma Y. and Chbeir R., Content and Structure Based Approach for XML Similarity, *CIT*, 136-140, 2005
15. Maguitman A. G., Menczer F., Roinestad H. and Vespignani A., Algorithmic Detection of Semantic Similarity. In *Proceedings of the 14th International WWW Conference*, 107-116, Japan, 2005.
16. Miller G., WordNet: An On-Line Lexical Database. *Int. Journal of Lexicography*, 1990.

17. Nierman A. and Jagadish H. V., Evaluating structural similarity in XML documents. In *Proceedings of the 5th International Workshop on the Web and Databases,* 2002.
18. Rada R., Mili H., Bicknell E. and Blettner M., Development and Application of a Metric on Semantic Nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19:17-30, 1989.
19. Ray E.T., Introduction à XML. *Edition O'Reilly, Paris, 327 p.,* 2001
20. Resnik P., Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proc. of the 14th IJCA-95*, Vol. 1, 448-453, Montreal, Canada, 1995.
21. Richardson R. and Smeaton A.F., Using WordNet in a Knowledge-based approach to information retrieval. In *Proceedings of the 17th Colloquium on Information Retrieval*, 1995.
22. Sanz I., Mesiti M., Guerrini G. and Berlanga Lavori R., Approximate Subtree Identification in Heterogeneous XML Documents Collections. *XSym*, 192-206, 2005.
23. Shasha D. and Zhang K., Approximate Tree Pattern Matching. In *Pattern Matching in Strings, Trees and Arrays*, chapter 14, Oxford University Press, 1995.
24. Tversky A., Features of Similarity. *Psychological Review*, 84(4):327-352, 1977
25. Wagner J. and Fisher M., The String-to-String correction problem. *Journal of the Association of Computing Machinery*, 21(1):168-173, 1974.
26. Wong C. and Chandra A., Bounds for the String Editing Problem. *Journal of the Association for Computing Machinery*, 23(1):13-16, January 1976.
27. WWW Consortium, The Document Object Model, http://www.w3.org/DOM.
28. Zhang K. and Shasha D., Simple Fast Algorithms for the Editing Distance Between Trees and Related Problems. *SIAM Journal of Computing*, 18(6):1245-1262, 1989.
29. Zhang Z., Li R., Cao S. and Zhu Y., Similarity Metric in XML documents. *Knowledge Management and Experience Management Workshop*, 2003.