ROMAI J., **4**, 2(2008), 61–72

THE SIMILARITY OF XML-BASED DOCUMENTS IN FINDING THE LEGAL INFORMATION

Sorina Cornoiu

Legislative Council, Bucharest

sorina_c_2000@yahoo.com

Abstract In this paper, we propose to integrate semantic similarity assessment in an edit distance algorithm, seeking to amend similarity judgments when comparing XML-based legal documents[3].

Keywords: information retrieval, XML, semantic, legal information.

2000 MSC: 68P20, 05C05, 03D55, 68Q55.

1. INTRODUCTION

In the legal information retrieval systems, the information is usually searched by means of a full text search, every term in the texts of the documents can function as a search key [4].

In the past few years, XML has been established as an effective mean for information management, and has been widely exploited for complex data representation[1]. We propose developing efficient techniques for comparing XML-based legal documents to become essential in information retrieval (IR) research, integrating IR semantic similarity assessment in an edit distance algorithm, seeking to amend similarity judgments when comparing XML-based legal documents. Our approach consists of an original edit distance operation cost model, introducing semantic relatedness of XML element/attribute labels, in traditional edit distance computations[1].

In recent years, W3C's XML (eXtensible Mark-up Language) has been accepted as a major mean 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 com-

pare XML-based legal documents becomes crucial in information retrieval [3]. Legal documents play an important role in all activities related to the legal domain. In particular they represent an efficient human communication mean to transmit legal knowledge. Legislations are often complex and prone to change. Organizations that base their daily work on a set of legal documents have to deal with a massive amount of legal and numerous legal updates[7]. Legal documents typically combine structured and unstructured information. The structured information is increasingly tagged with markup languages such as XML (Extensible Markup Language) [6] . 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 legal information retrieval systems, estimating semantic similarity between legal documents is of key importance to improving search results[1].

Semantic similarity IR for legal information, incited us to expand existing XML structural similarity so as to take into account semantic relatedness while comparing XML documents [1].

2. BACKGROUND

2.1. LEGAL XML DATA MODEL

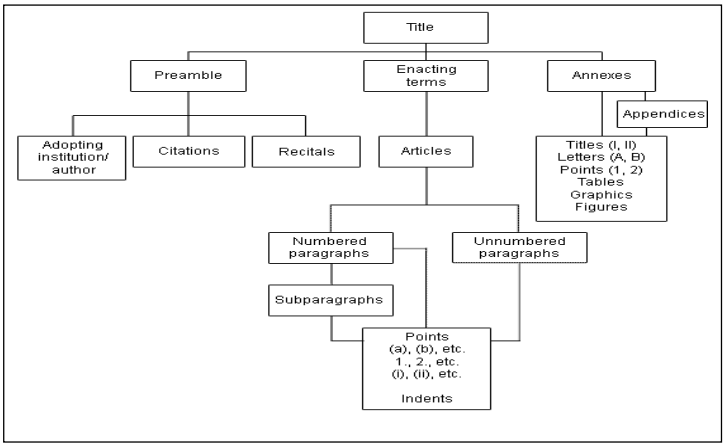
Legal documents typically combine structured and unstructured information, the former, for instance, referring to common document architectures, reference structures and metadata information the latter involving the natural language texts. The structured information is increasingly tagged with markup languages such as XML [6].

XML documents represent a hierarchically structured information and can be modeled as Ordered Labeled Trees (OLTs). 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 [1].

The Community Official Journal texts, which constitute our working base, consist of many types of texts grouped in two main categories: legislation, information and notices. In this paper, we focus on regulations, directives, decisions and recommendations regardless of their category [5]. Legislative documents have a hierarchical structure in which elements with detailed content are nested in larger elements[1]. In [4] we can find recommendations and legislative techniques for structuring the document.

Community acts are generally drafted according to a standard structure (fig. 1):

Fig. 1. Basic structure of legislative acts



1. the "Title" comprises all the information in the heading of the act which serves to identify it. It may be followed by certain technical data (reference to the authentic language version, relevance for the EEA, serial number) which are inserted, where appropriate, between the title proper and the preamble [4];

2. "preamble" means everything between the title and the enacting terms of the act, namely the citations, the recitals and the solemn forms which precede and follow them [4]. Citations: at the beginning of the preamble, they indicate the legal basis of the act, the proposals, recommendations, initiatives, drafts... that must be obtained, and certain opinions and other non-mandatory pro-

cedural steps. Citations are generally introduced by the dedicated expression 'Having regard to" or "Acting in accordance with" [5]. Recitals: are the parts of the act containing the statement of reasons for the act; they are placed between the citations and the enacting terms. Recitals are introduced by the word "Whereas : " and continue with numbered points comprising one or more complete sentences [5];

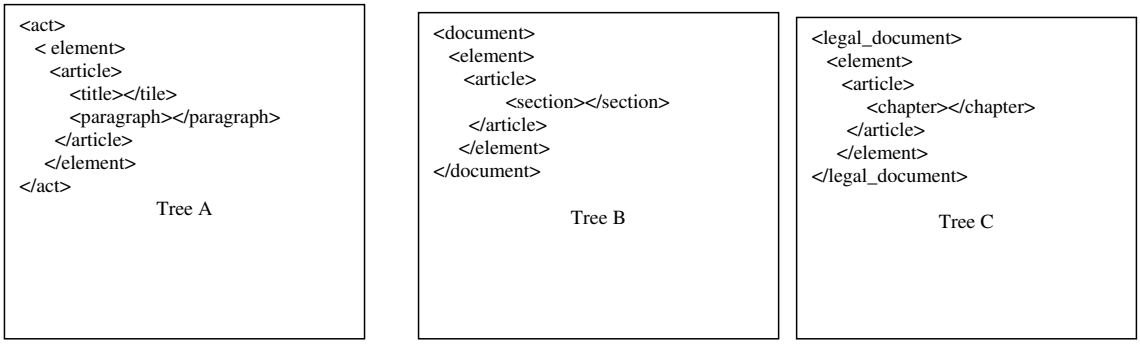
3. the "enacting terms" are the legislative part of the act. They are composed of articles, which may be grouped in titles, chapters and sections, and may be accompanied by annexes [5].

4. annex: "where necessary annex" and is spread out until the end of document. In case where many annexes are necessary, each annex has a heading like the one cited above and is numbered [5].

2.2. XML STRUCTURAL SIMILARITY

An XML document can be modeled as an ordered labeled tree, each node in the tree corresponding to an element in the document and is labeled with the element tag name. Let us consider the following XML documents (fig. 2).

Fig. 2. Example 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. However, despite having similar structural characteristics, one can obviously

recognize that sample document A shares more semantic characteristics with document B than with C. For example, in fig. 2 pairs act-document and paragraph-section, from documents A and B, are semantically similar while paragraph-chapter, from documents A and C, are semantically different [3]. For this reason 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 [1].

In order to determine structural similarities between hierarchically structured data, particularly XML documents, we can use various methods : Tree Edit Distance (TED) Similarity, Tag Similarity, Fourier Transform Similarity Metric.

Several authors have provided algorithms for computing the optimal edit distance between two trees. In general, the edit distance measures the minimum number of node insertions, deletions, and updates required to convert one tree into another [2]

$$TED(D_i, D_j) = \frac{editDist(D_i, D_j)}{max(|N_i|, |N_j|)}, \tag{1}$$

where : N_i is the set of nodes in the tree representation of document D_i and N_j is the set of nodes in the tree representation of document D_j. Chawathe restricts insertion and deletion operations to leaf nodes, and changes to trees using three basic tree edit operations : insertion of leaf nodes - Ins(x, i, p, λ(x)) , deletion of leaf nodes - Del(x, p) , update internal/leaf nodes - Upd(x, y) [3] . By associating costs with each edit operation, Chawathe defines the cost of an edit script (sequence of edit operations) to be the sum of the costs of its component operations [1]. Similarity measures based on edit (or metric) distance are generally computed as [3]

$$sim(A, B) = \frac{1}{1 + dist(A, B)}. \tag{2}$$

We can to find the cheapest sequence of edit operations that can transform one tree into another.

Chawathe employed the ld-pair representation of a tree node. It is defined as the pair (l, d) where: l and d are the node label and depth in the tree [3], respectively. We use p.l and p.d to refer to the label and the depth of an ld-pair p respectively. For example, we have the following representation for the tree A :

A=((act,0),(enact,1),(article,2),(title,3),(paragraph,3))

We can assign identical costs to insertion and deletion operations (CostIns = CostDel = 1), as well as to update operations only when the newly assigned label is different from the node current label (CostUpd(a ,b) = 1 when a.l $\neq$ b.l, otherwise, when the labels are the same, CostUpd = 0, underlining that no changes are to be made to the label of node a).

Applying Chawathe's approach we obtain

Edit script= Upd (A[1],B[1]) , Upd (A[4],B[4]) , Del (A[5],A[3])
Dist(A,B)=Dist(A,C)=3.

Using (2) , we obtain : Sim(A, B) = Sim(A, C) = 0.25.

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.

Table 1 Computing minimum edit distance for XML trees A and B

	0	B[1]/(document,0)	B[2]/(element,1)	B[3]/(article,2)	B[4]/(section,3)
0	0	1	2	3	4
A[1](act,0)	1	1	2	3	4
A[2](element,1)	2	2	1	2	3
A[3](article,2)	3	3	2	1	2
A[4](title,3)	4	4	3	2	2
A[5](paragraph,3)	5	5	4	3	3

Intuitive cost models do not affect the correctness of Chawathe's structural similarity algorithm. However, they fail to capture the semantics of XML doc-

uments. we propose to complement the structure-based similarity algorithm, with a cost model integrating semantic assessment (semantic similarity) in the comparison process [1].

So far we spoke about how we can enhance existing XML comparison approaches in order to take into consideration both structural and semantic characteristics of XML documents.

3. INTEGRATED SEMANTIC AND STRUCTURE BASED SIMILARITY

In order to take into account semantic meaning while comparing XML documents, we propose to complement Chawathe's edit distance algorithm with the following semantic cost model (SCM)

$$Cost_{Op}(x, y) = Cost_{SemOp}(x, y) \times Cost_{DepthOp}(x) \in [0, 1] \quad (3)$$

where Op designates an insertion, deletion or update operation, Cost_{SemOp}(x, y) is the label semantic similarity cost and represents the operation costs according to the semantic of concerned nodes, $\text{Cost}_{DepthOp}$(x) represents the node depth.

In our case, using (3) we have

Cost_{Op}(A,B)= Cost_{SemOp}(A,B) × $\text{Cost}_{DepthOp}$(A)

3.1. LABEL SEMANTIC SIMILARITY COST

The label semantic similarity cost Cost_{SemOp}(x, y) represents the operation costs according to the semantic of the concerned nodes

$$Cost_{SemOp}(x, y) = 1 - Sim_{SemOp}(x.l, y.l) \quad (4)$$

where Sim_{SemOp}(x.l, y.l) represents the label semantic similarity that use Lin's semantic similarity measure.

For calculate Lin's measure WordNet-based semantic similarity. The WordNet (WordNet Search - 3.0) is a lexical database that provides a combination between the traditional lexicographical information and modern computing. WordNet contains more than 118000 different word forms and more than 90000 different word senses and include synonymy (same-name) , antonymy

(opposite-name), hyponymy (sub-name), hypernymy (super-name), meronymy (part-name) and holonymy (whole-name) relations. The lexical database WordNet is particularly suited for similarity measures, since it organizes nouns and verbs into hierarchies of "is-a" relations. Using the CPAN(Comprehensive Perl Archive Network) module, we are able to measure the semantic similarities between words by use of algorithms. There will be used the Lin's algorithm that is based on the information content of the least common subsumer (LCS) of concepts A and B. Information content is a measure of the specificity of a concept, and the LCS of concepts A and B is the most specific concept that is an ancestor of both A and B. The Lin measure augments the information content of the LCS with the sum of the information content of concepts A and B themselves. The Lin measure scales the information content of the LCS by this sum [8].

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

$$Sim_{Sem}(w1, w2) = Sim_{Sem}(c1, c2) = \frac{2logp(c_0)}{logp(c_1) + logp(c_2)} \tag{5}$$

where:
- c1 and c2 are concepts, in a knowledge base of hierarchical structure (taxonomy), subsuming words w1 and w2 respectively;
- c0 is the most specific common ancestor of concepts c1 and c2;
- 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) = Σcount(w) and N: total number of words in the corpus.

We can have the following three situations:

$$update : Cost_{SemUpd}(x, y) = 1 - Sim_{Sem}(x.l, y.l) \tag{6}$$

where x is an node from an XML document and x.l is the node label x, y is an node from an XML document and y.l is the node label y.

When labels are identical, the 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, the semantic similarity is of minimum

value, Sim_{Sem}(x.l, y.l) = 0, which brings us to Cost_{Upd}(x, y) = 1 [3];

$$insert : Cost_{SemIns}(x, i, p, \lambda(x)) = 1 - Sim_{Sem}(\lambda(x), p.l) \tag{7}$$

$$delete : Cost_{SemDel}(x, p) = 1 - Sim_{Sem}(x.l, p.l). \tag{8}$$

In order to insert deletion operation, when labels are identical or completely different, insertion/deletion costs would be equal to 0 or 1 [3].

3.2. NODE DEPTH COST

Information becomes increasingly specific as one descends in the XML tree hierarchy. For example, consider the XML sample tree A in fig. 2. Editing node A[1] (A[1].l = act) by changing its label to "book", would semantically affect tree A a lot more than deleting node A[4] (A[4].l = title), 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, could be used for that matter [1]

$$Cost_{DepthOp}(x) = \frac{1}{1 + x.d} \in [0, 1] \tag{9}$$

where Op is an insert, delete or update operation, x.d is the depth of node considered for insertion, deletion or updating.
Editing the root node of a document tree involves $\text{Cost}_{DepthOp}$(root) = 1.

3.3. SEMANTIC COST MODEL(SCM)

In order to enrich formula (2) with semantic meaning, we propose the following cost model [1]

$$Cost_{Op}(x, y) = Cost_{SemOp} \times Cost_{Depth}(x) \tag{10}$$

where Op designates an insertion, updating or deletion operation. In this case we have

$$sim(A, B) = \frac{1}{1 + distSCM(A, B)}. \tag{11}$$

First we must obtain word semantic similarities, computed by following Lin's measure (Table 2).

Table 2 Word semantic similarities, computed by following Lin's measure.

Word pairs		SimLin	Wordpairs		SimLin
act	document	0.8391	article	chapter	0.3693
act	legal_document	0.8746	title	paragraph	0.5029
act	element	0.1067	title	section	06028
act	article	0.3768	title	chapter	0.5642
act	title	0.5758	title	document	0.6637
act	paragraph	0.5229	title	legal_document	0.6444
act	section	0.6316	paragraph	document	0.5944
act	chapter	0.5895	paragraph	section	0.5451
element	document	0.1252	paragraph	legal_document	0.5789
element	legal_document	0.1211	paragraph	chapter	0.5134
element	article	0.1017	chapter	legal_document	0.6616
element	section	0.1123	chapter	document	0.682
element	chapter	0.1043	section	legal_document	0.7152
element	title	0.1017	document	legal_document	0.9637
element	paragraph	0.0918	document	section	0.7391
article	legal_document	0.4218	document	article	0.4344
article	title	0.3611	article	section	0.3945
article	paragraph	0.3292			

The results attained by applying the semantic cost model to compare sample XML documents A, B and C are shown in Tables 3 and 4.

For example, $\text{Dist}_{SCM(A[1],B[1])}$= (1-SimLin(A[1], B[1])) = 1-0.8391=0.1609.

Table 3 Computing minimum edit distance for XML trees A and B; $\text{dist}_{SCM(A,B)}$=0.1014.

	B[1]/(document,0)	B[2]/(element,1)	B[3]/(article,2)	B[4]/(section,3)
A[1](act,0)	0.1609	0.8933	0.6232	0.3684
A[2](element,1)	0.4374	0.1609	04491	0.4438
A[3](article,2)	0.1885	0.2994	0.1609	0.2018
A[4](title,3)	0.0840	0.2245	0.1597	0.0993
A[5](paragraph,3)	0.1014	0.2270	0.1677	0.1137

Table 4 Computing minimum edit distance for XML trees A and C; $\text{dist}_{SCM(A,C)}$=0.1254.

	C[1]/(legal_document,0)	C[2]/(element,1)	C[3]/(article,2)	C[4]/(chapter,3)
A[1](act,0)	0.1254	0.8933	0.6232	0.4105
A[2](element,1)	0.4374	0.1254	04491	0.4438
A[3](article,2)	0.1927	0.2994	0.1254	0.2102
A[4](title,3)	0.0889	0.2245	0.1597	0.1089
A[5](paragraph,3)	0.1052	0.2270	0.1677	0.1216

By using (11), we obtain: $\text{Sim}_{SCM(A,B)} = 1/(1+0.1014)= 0.907$ and $\text{Sim}_{SCM(A,C)} = 1/(1+0.1254)= 0.888$.

Considering semantic relatedness, in the comparison process, reflects the fact that sample documents A and B are more similar than A and C (SimSCM(A, B) > SimSCM(A,C)), in spite of sharing identical structural similarities [3]. In order to compare XML legal documents we use a structure-based (edit distance) similarity algorithm, which seems to capture semantic meaning effectively.

4. CONCLUSION

The database community has proposed several languages for quering XML, such as XML-QL, XQuery, XQL. These language are based on exact matching and do not support ranked queries. For this reason we propose using semantic cost model which takes in consideration both structural and semantic characteristics of XML legal documents.

References

[1] J. Tekli, R. Chbeir, K. Yétongnon, *A hybrid approach for XML similarity*, 33rd International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)pringer Verlag Incs, Harrachov, Czech Republic, January 2007, 783-795.

[2] D. Buttler, *A short survey of document structure similarity algorithms*, The 5th International Conference on Internet Computing, Las Vegas, NV, United States, June 21, 2004.

[3] J. Tekli, R. Chbeir, K. Yétongnon, *Semantic and Structure Based XML Similarity-AnIntegratedApproach*, 13th International Conference on Management of Data (COMAD'06), Delhi, India, December 2006.

[4] *Joint practical guide for the drafting of community legislation*, http://reterei.eu/rete/GPC_en.pdf.

[5] F. Fady, R. Rousselot, *DARES: Documents annotation and recombining system-Application to the European law*, Artificial Intelligence and Law, Springer, **15**,(2)(2007), 83-102.

[6] M.F. Moens, *Retrieval of legal documents: combining structured and unstructured information*, Proceedings ELPUB2005 Conference on Electronic Publishing, Kath. Univ. Leuven, June 2005.

[7] O. Vasutiu, D. Jouve, Y. Amghar, J.M. Pinon, *XML based legal document drafting information system*, Standards for Legislative XML Workshop, Amsterdam, December 15, 2007.

[8] T. Pedersen, J. Michelizzi , *Wordnet: Similarity - measuring the relatedness of concepts*, Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04), 2004.