경희대학교 KYUNG HEE UNIVERSITY

Ubiquitous Computing Laboratory
Kyung Hee University, Korea

Presentation of the Ph.D. thesis entitled

XML Data Integration and Transformation – An Attempt to Enhance Data Sharing between Applications

by Pham Thi Thu Thuy

Advisor: Prof. Young-Koo Lee

Co-advisor: Prof. Sungyoung Lee

April 28th 2012

Outline

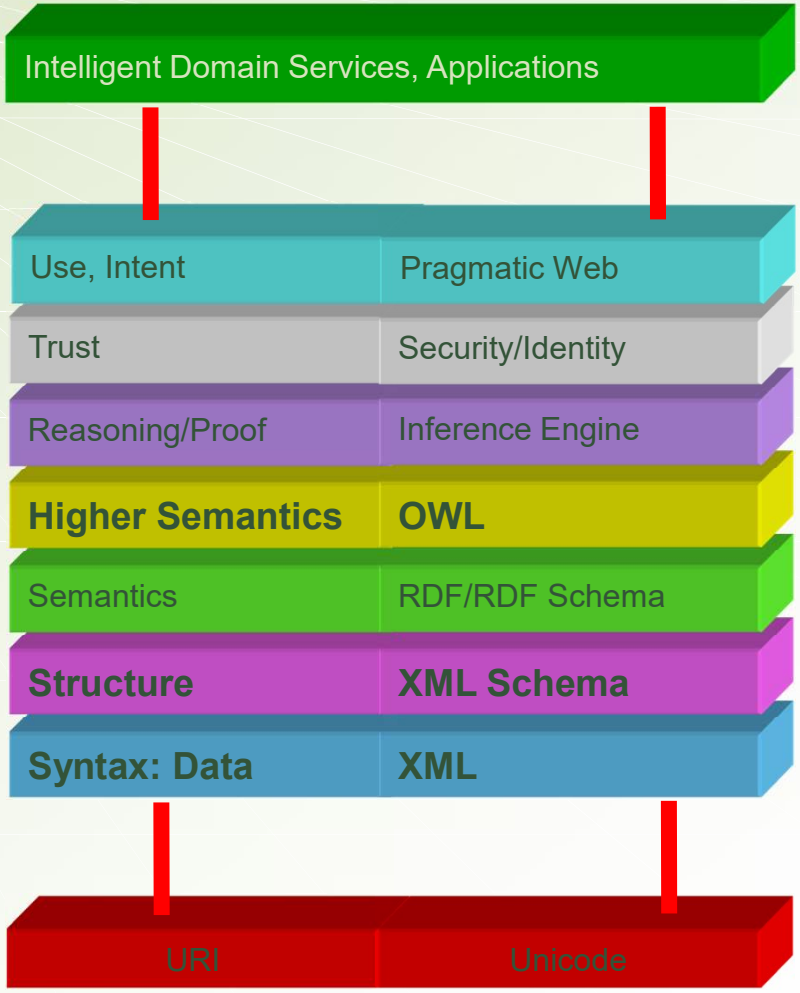
1. Introduction
2. Related Work
3. Thesis Contribution
4. ESim: A Method of XML Integration
5. S-Trans: XML Transformation
6. Experiments
7. Conclusions and Future Work

Introduction

- XML (*eXtensible Markup Language*): common data representation format - a standard for sharing data.
- Web-based applications and services publish their data using XML.
- Heterogeneity problem: Same information can be published using XML in different structures and terminology.
  - Sharing XML is not yet fully automatic.
- Problem solutions: schema matching, schema integration, and schema transformation in the context of XML data.
- Algorithms which automate these tasks will reduce time and efforts spent on creating and maintaining data sharing between applications: e-business, e-science, e-learning, etc.

XML in the Semantic Web stack

- XML: Focus on syntax of data.
- XSD (*XML Schema Definition*) : used to describe the structure of XML document.
- OWL (*Web Ontology Language*): support for data semantic - add more vocabulary for describing properties, classes, relations between classes, cardinality, equality, etc.



Semantic Web Stack, from Tim Berners-Lee presentation for Japan Prize, 2002

Term Definitions

According to Business dictionary:
http://www.businessdictionary.com/definition/
Angielski online (*):
http://www.tlumaczenia-angielski.info/linguistics/semantics.htm

Structure

- Construction of identifiable elements.
- Each element is functionally connected to others.
- Element's interrelationships are fixed or changing occasionally.

Semantics

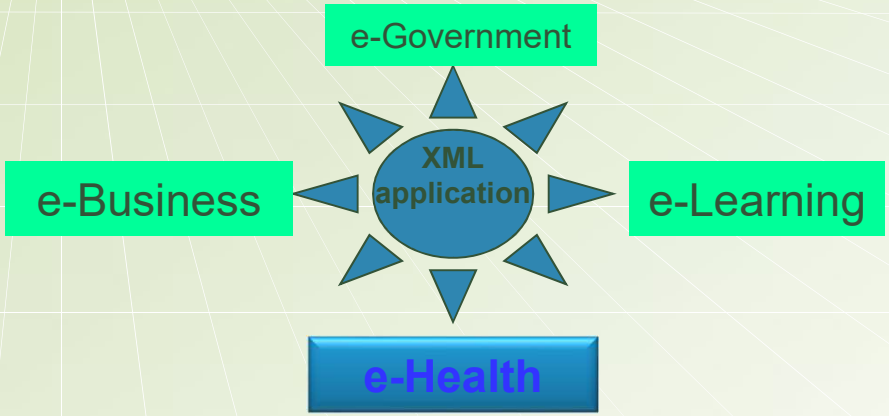
- Scientific study of the meaning of words (*).
- meaning is analyzed in terms of their semantic features: the way in which words are used in document.

Implicit semantics

- Semantics that are hidden in the document .
- E.g. Data type, cardinality constraint, etc.

Thesis Motivation

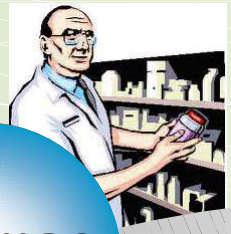
- Since the growing number of XML data in various applications and heterogeneity of XML:



- There is a need to enhance data sharing between applications. This thesis focuses on two methods:
  - Integrate similar XML data into single integrated source (ESim).
  - Transform XML data into higher semantic supporting language, e.g. OWL (S-Trans).

Example “XML data integration”

Reduce time and efforts spent on exchanging patient’s information



Pharmacy

Doctor

Hospital

Integrated XML source

Specialist clinic

Insurance company

Patient’s information is stored by various healthcare services in XML format.

ESim - Problem statements

Similar XML contents are described by different names & structures.

To integrate XML data, accurate similarity measure is necessary.

The structure & implicit semantics of XML document are represented through XSD.

- Since XML contains structure ➔ easy to measure structural similarity.
- But, how to measure semantic similarity of XML documents?

S-Trans: Problem & Motivation

XML disadvantage

Lack of description: problems when semantic agents want to understand and reason about XML data.

XML2OWL

XML2OWL: data redundancy on duplicate XML.
Resolution: How to transform XML duplicate elements into OWL ontology.

S-Trans

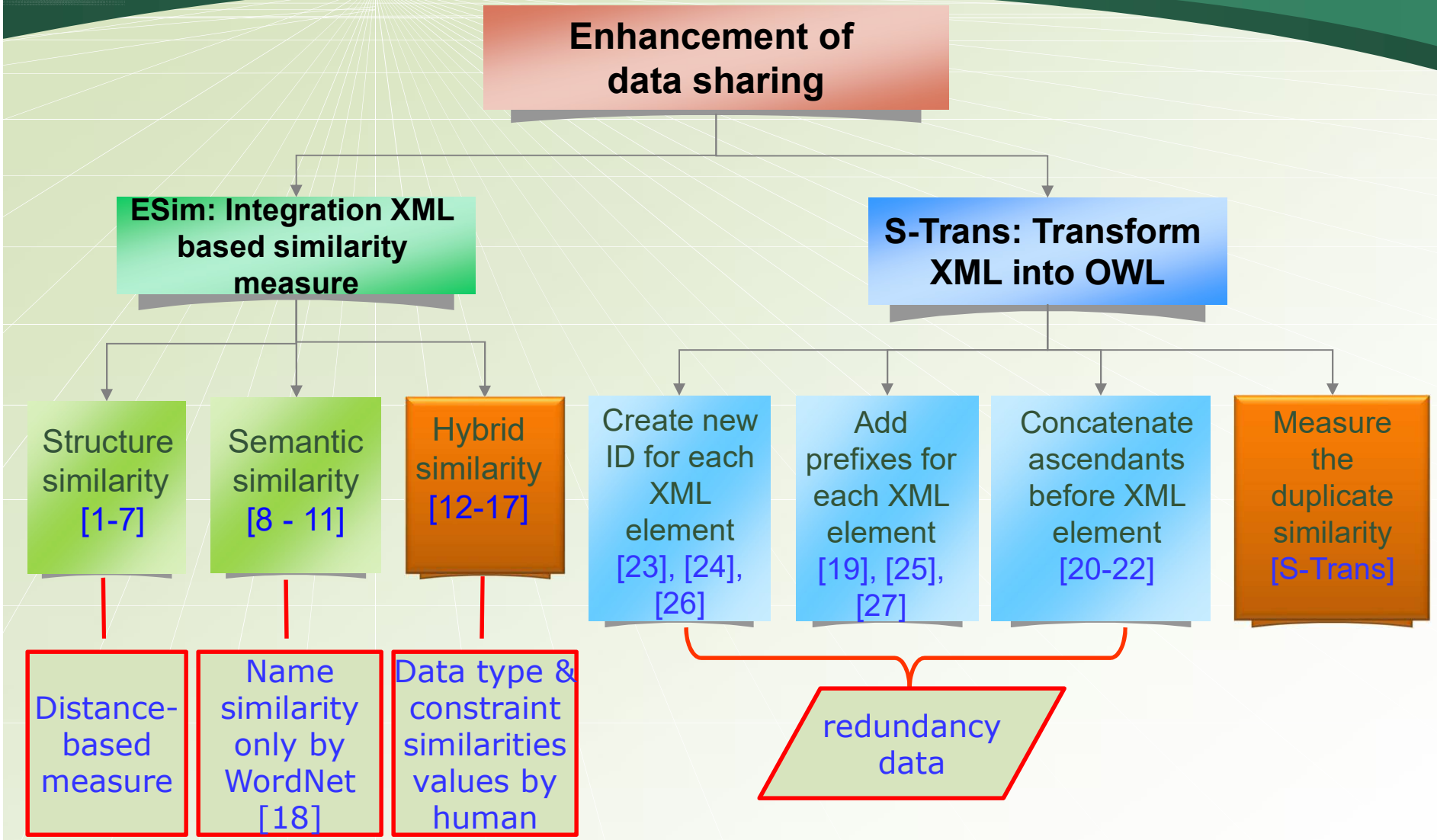
Duplicate similarity measure & transformation : novel semantic similarity measure.

```
<GunLicense>
  <registeredGun>
    <Gun>
      <serial>ABCD</serial>
    </Gun>
  </registeredGun>
  <holder>
    <Person>
      <driversLicenseNumber>ZXYZXY</driversLicenseNumber>
    </Person>
  </holder>
</GunLicense>
```

1 **How many guns/people are registered in a gun license?**

2 **Can this gun be registered in other gun licences?**

Related work



Related work (cont')

Hybrid Similarity [12-17]

Wojnar et al. [13], [14]

- ✓ Structure: reuse tree distance based method
- ✓ Semantic = string similarity + Cardinality constraint
- ➢ manually given

Kim et al. [16], Song et al. [17]

- ✓ Path matching of two XMLs based on Cosine metric.

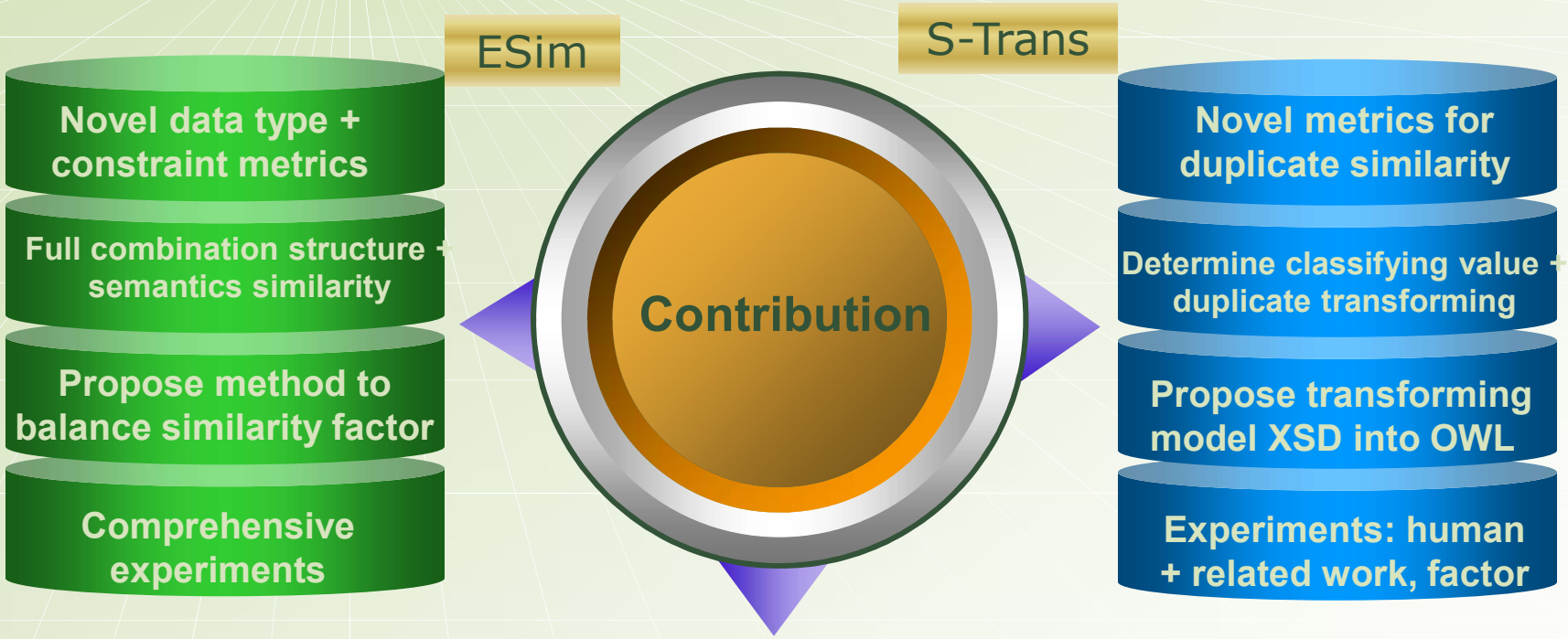
$$sim(d_i, d_j) = \frac{D_i . D_j}{|D_i| \times |D_j|} = \frac{\sum_{k=1}^{t} w_{ki} \times w_{kj}}{\sqrt{\sum_{k=1}^{t} w_{ki}^2} \times \sqrt{\sum_{k=1}^{t} w_{kj}^2}}$$

- ➢ focus mostly on structure

Nayak et al. [12], Algergawy et al. [15]

- ✓ Structure: neighborhood relationships
- ✓ Semantic = name + Data type + Cardinality constraint
- ➢ manually given

Thesis contribution



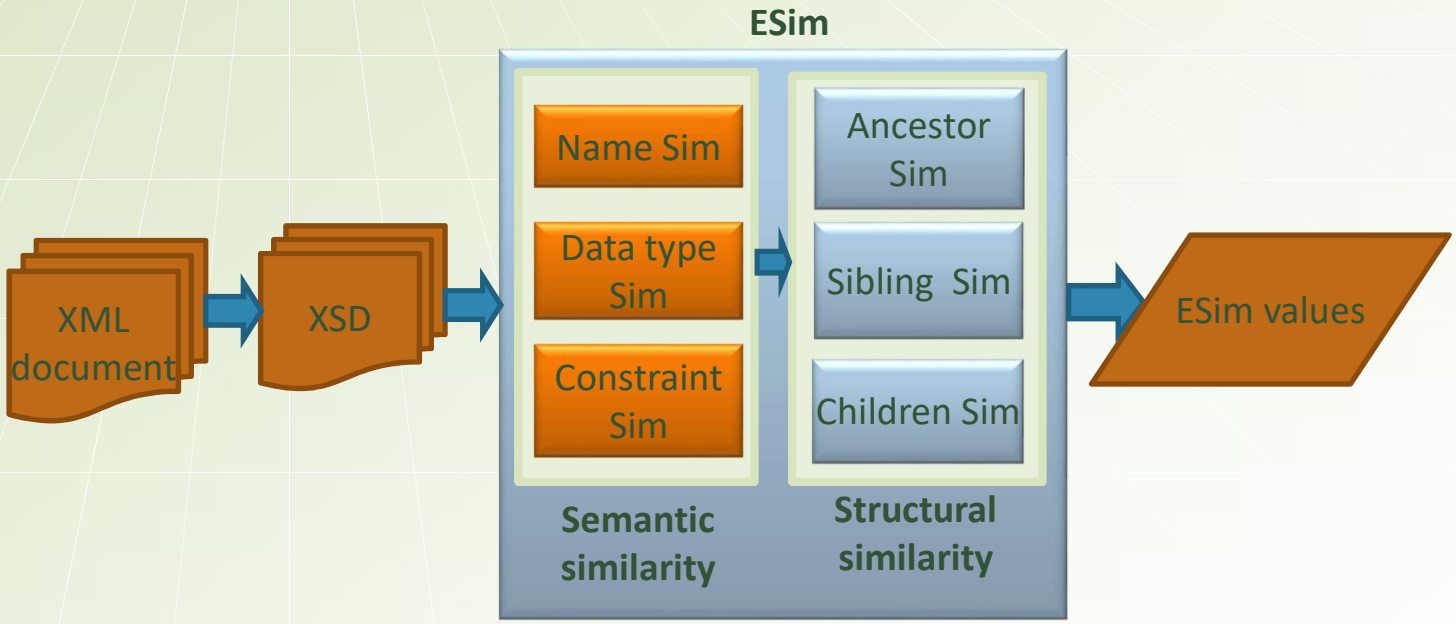
Two methods for enhancing the data sharing between applications.

ESim: Element Similarity Measure for Integration of XML data

Pham Thi Thu Thuy, Young-Koo Lee, and Sungyoung Lee, "Semantic and Structural Similarity Analysis for Integration of Ubiquitous XML Healthcare Data", Journal of Personal and Ubiquitous Computing (SCIE, IF: 1.137), ISSN: 1617-4909, 2012.

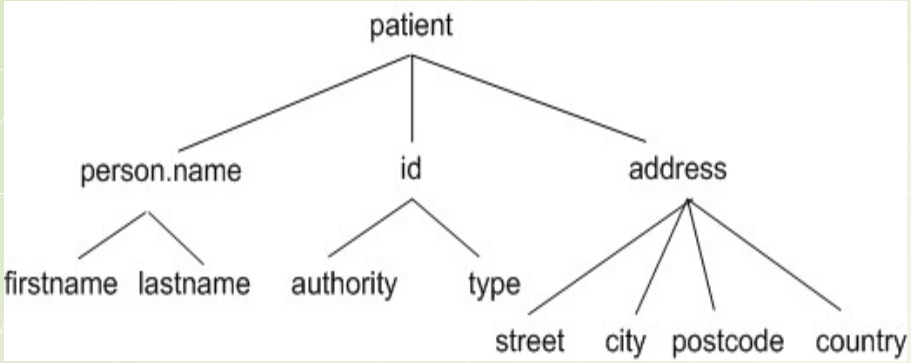
ESim framework

- Framework includes:
  - Input: XSD document (may be extracted by HIT soft. [23])
  - ESim computation: Semantic & Structure similarity measures
  - Output: ESim similarity values.

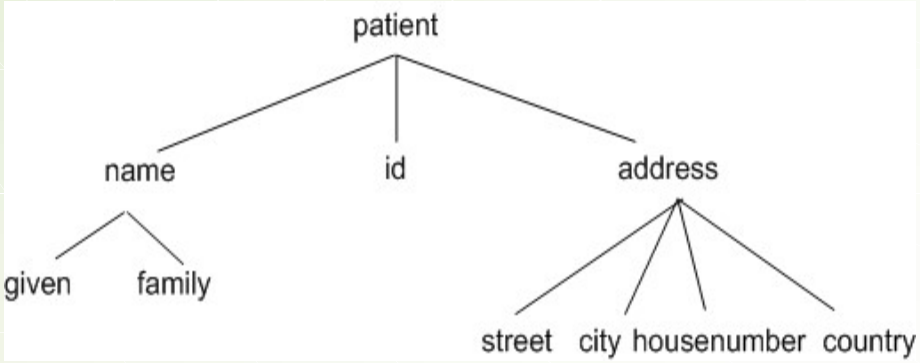


Motivating example

- Find the similarity between elements in schema Patient_A and those in schema Patient_B.



Tree representation for Schema Patient_A



Tree representation for Schema Patient_B

ESim Similarity

- The similarity (*ESim*) between two elements e_1 and e_2 in two different XSDs.
- The weighted sum of semantic similarity (*SeSim*) and structural similarity *(StSim):*

$$ESim(e_1, e_2) = \alpha * SeSim(e_1, e_2) + (1 - \alpha) * StSim(e_1, e_2) \quad (1)$$

where α is the weighted value.

- According to experiments, ESim has close value to user's perspective at α=0.55

Semantic Similarity

- Semantic features of XML: element names and their descriptions.
- ***Semantic similarity*** (*SeSim*) captures the similarities of element names *(NSim)*, data type *(DSim)*, and constraints *(CSim)* of two elements.

$$SeSim(e_1, e_2) = \beta * NSim(e_1, e_2) + \chi * DSim(d_1, d_2) + (1 - \beta - \chi) * CSim(e_1, e_2) \quad (2)$$

where β and χ are the weighted constants.

- According to experiment, at the values of β=*0.4*, χ=*0.3*, similarity score proposed by our method returns close results to user's perspective.

Name Similarity [8, 12, 15]

- Measure the meaning of element name.
- Reuse distance based measure [29] to compute the semantic similarity of e_1 in XSD_1 and e_2 in XSD_2 by referring them in WordNet [18]:

$$NSim(e_1, e_2) = \frac{2 * depth(LCS)}{depth(e_1) + depth(e_2)} \quad (3)$$

where *depth(LCS)* is # nodes from the common super-concept of e_1 and e_2 to the root node; *depth(e_1)* and *depth(e_2)* are # nodes from e_1 and e_2 to the root node.

A fragment of WordNet

- E.g.

$$NSim(lecturer, professor) = \frac{2 * 4}{5 + 6} = 0.73$$

Name Similarity (cont')

- If compared elements are combination of words → tokenized them before measuring their similarity in WordNet, then compute by (4) or (5). :

$$NSim(E_1, E_2) = \frac{\sum_{i=1}^{m} \max_{j=1}^{n}(NSim(e_{1_i}, e_{2_j}))}{m}, \quad m \geq n \qquad (4)$$

$$NSim(E_2, E_1) = \frac{\sum_{i=1}^{n} \max_{j=1}^{m}(NSim(e_{2_i}, e_{1_j}))}{n}, \quad n > m \qquad (5)$$

where *m* and *n* are # tokenized words of element E_1 and E_2, respectively.

- Some cases, element names are not in WordNet ➔ define metric for measuring string similarity:

$$NSim(e_1, e_2) = LingSim(e_1, e_2) = \frac{n_{e_1 \cap e_2}}{\max(n_{e_1}, n_{e_2})} \qquad (6)$$

where $n_{e_1 \cap e_2}$ is # matching characters between elements e_1 and e_2; *max* is the maximum value; n_{e_1} and n_{e_2} are the lengths of the elements e_1 and e_2, respectively.

Data type Similarity

- Data type is one of the semantic features of elements.
- Novel measure: Data type similarity of two element e_1 and e_2 is the fraction of common number constraining facets [23] per the maximum *(max)* number of constraining facets of each element :

$$DSim_1(e_1, e_2) = \frac{\sum_i \left| \{ cf_i \mid e_1[cf_i] = e_2[cf_i], 1 \le i \le n_{cf} \} \right|}{max(\# e_1.cf, \# e_2.cf)} \quad (7)$$

- To improve (7) $\rightarrow$ equation (8)

$$DSim_2(e_1, e_2) = \frac{max(\# e_1.cf, \# e_2.cf)}{n_{cf}} \quad (8)$$

where n_{cf} is the number of constraining facets.

- Data type similarity (*DSim*) is the weighted function of equations (7) and (8).

$$DSim(e_1, e_2) = \frac{\delta_1 * DSim_1(e_1, e_2) + \delta_2 * DSim_2(e_1, e_2)}{\delta_1 + \delta_2} \quad (9)$$

Data type Similarity (cont')

- Text= {length, minLength, maxLength, pattern, enumeration, whiteSpace}
- Date time (Dtime)= {pattern, enumeration, whiteSpace, minInclisive, maxInclusive, minExclusive, maxExclusive}

$$DSim_1(e_1, e_2) = \frac{3}{7} = 0.43$$

- Since there are 12 constraining facets:

$$DSim_2(e_1, e_2) = \frac{7}{12} = 0.58$$

- Assume that $DSim_1$ and $DSim_2$ have similar roles, $\delta_1 = \delta_2 = 0.5$.

$$DSim(e_1, e_2) = \frac{0.5 * 0.43 + 0.5 * 0.58}{0.5 + 0.5} = 0.51$$

Data Type Similarity (cont')

Similarity values of data types resulting from equation (9):

	URI	lang	text	Ubyte	dec	int	Dtime	Name	Entity	ID	Token	Type
URI	1.00	0.43	0.50	0.28	0.51	0.31	0.51	0.43	0.36	0.36	0.51	0.51
lang	0.43	1.00	0.43	0.28	0.31	0.27	0.31	0.39	0.33	0.33	0.35	0.32
text	0.50	0.43	1.00	0.33	0.51	0.31	0.51	0.43	0.36	0.36	0.39	0.41
Ubyte	0.28	0.28	0.33	1.00	0.33	0.53	0.49	0.28	0.22	0.22	0.24	0.23
dec	0.51	0.31	0.51	0.33	1.00	0.56	0.54	0.31	0.24	0.24	0.27	0.28
int	0.31	0.27	0.31	0.53	0.56	1.00	0.47	0.27	0.21	0.21	0.23	0.21
Dtime	0.51	0.31	0.51	0.49	0.54	0.47	1.00	0.31	0.24	0.24	0.27	0.28
Name	0.43	0.39	0.43	0.28	0.31	0.27	0.31	1.00	0.33	0.33	0.35	0.33
Entity	0.36	0.33	0.36	0.22	0.24	0.21	0.24	0.33	1.00	0.32	0.33	0.31
ID	0.36	0.33	0.36	0.22	0.24	0.21	0.24	0.33	0.32	1.00	0.33	0.31
Token	0.51	0.35	0.39	0.24	0.27	0.23	0.27	0.35	0.33	0.33	1.00	0.33
Type	0.51	0.32	0.41	0.23	0.28	0.21	0.28	0.33	0.31	0.31	0.33	1.00

Cardinality Constraint Similarity

- Cardinality constraint (*minOccurs*, *maxOccurs*) is also one of the semantic features of element.
- Novel measure: For the definitely values of *minOccurs* and *maxOccurs*:

$$CSim(e_1(\min,\max),e_2(\min,\max)) = \frac{\left(1-\frac{|e_1.\min-e_2.\min|}{e_1.\min+e_2.\min}\right)+\left(1-\frac{|e_1.\max-e_2.\max|}{e_1.\max+e_2.\max}\right)}{2} \quad (10)$$

where *min*, and *max* are short forms of *minOccurs*, and *maxOccurs*, respectively.

- Usually, the value of *maxOccurs = unbound*. To measure the *CSim* for this value, we propose the following equation:

$$e_1[maxOccurs = unbound] = \frac{4294967296 + e_2[maxOccurs]}{2} \quad (11)$$

where *4294967296* is the maximum value declared for *maxOccurs* property, suggested by Microsoft [43].

Structural Similarity

- Measure the similarity of elements having functional connected with current elements.
- Structural similarity *(StSim)* between two elements e_1 and e_2 is defined as weighted sum of ancestor similarity *(AcSim)*, sibling similarity (SbSim), and children similarity (*ChSim*):

$$StSim(e_1, e_2) = \frac{\varepsilon_1 * AcSim(e_1, e_2) + \varepsilon_2 * SbSim(e_1, e_2) + \varepsilon_3 * ChSim(e_1, e_2)}{\varepsilon_1 + \varepsilon_2 + \varepsilon_3} \quad (12)$$

- The *AcSim*, *SbSim*, and *ChSim* are computed by collecting all their respective elements and then compared their semantic similarity.

Ancestor Similarity

- Assume that E_1 and E_2 are the collections of ancestor elements of e_1 and e_2, respectively:

$$AcSim(E_1, E_2) = \begin{bmatrix} SeSim(e_{1_1}, e_{2_1}) \cdots SeSim(e_{1_1}, e_{2_n}) \\ \vdots \quad \ddots \quad \vdots \\ SeSim(e_{1_m}, e_{2_1}) \cdots SeSim(e_{1_m}, e_{2_n}) \end{bmatrix}, m \geq n \qquad (13)$$

$$AcSim(E_2, E_1) = \begin{bmatrix} SeSim(e_{2_1}, e_{1_1}) \cdots SeSim(e_{2_1}, e_{1_m}) \\ \vdots \quad \ddots \quad \vdots \\ SeSim(e_{2_n}, e_{1_1}) \cdots SeSim(e_{2_n}, e_{1_m}) \end{bmatrix}, m < n \qquad (14)$$

where *m* and *n* are the total number of ancestor elements of e_1 and e_2, respectively

$$AcSim(E_1, E_2) = \frac{\sum_{i=1}^{m} \max_{j=1}^{n} (SeSim(e_{1_i}, e_{2_j}))}{m}, m \geq n \qquad (15)$$

$$AcSim(E_2, E_1) = \frac{\sum_{i=1}^{n} \max_{j=1}^{m} (SeSim(e_{2_i}, e_{1_j}))}{n}, m < n \qquad (16)$$

Sibling Similarity

- Assume that E_1 and E_2 are the collections of siblings of e_1 and e_2, respectively:

$$SbSim(E_1,E_2)=\begin{bmatrix} SeSim(e_{1_1},e_{2_1}) & \cdots & SeSim(e_{1_1},e_{2_n}) \\ \vdots & \ddots & \vdots \\ SeSim(e_{1_m},e_{2_1}) & \cdots & SeSim(e_{1_m},e_{2_n}) \end{bmatrix}, m \geq n \qquad (17)$$

$$SbSim(E_2,E_1)=\begin{bmatrix} SeSim(e_{2_1},e_{1_1}) & \cdots & SeSim(e_{2_1},e_{1_m}) \\ \vdots & \ddots & \vdots \\ SeSim(e_{2_n},e_{1_1}) & \cdots & SeSim(e_{2_n},e_{1_m}) \end{bmatrix}, m < n \qquad (18)$$

where *m* and *n* are the total number of sibling elements of e_1 and e_2, respectively

$$SbSim(E_1,E_2)=\frac{\sum_{i=1}^{m}\max_{j=1}^{n}(SeSim(e_{1_i},e_{2_j}))}{m}, m \geq n \qquad (19)$$

$$SbSim(E_2,E_1)=\frac{\sum_{i=1}^{n}\max_{j=1}^{m}(SeSim(e_{2_i},e_{1_j}))}{n}, m < n \qquad (20)$$

Children Similarity

- Measure the similarity of all immediate children.
- Find the path matching pairs between two elements.
- Take the average similarity value.
- The children similarity between two duplicates *e1* and *e2* is specified as:

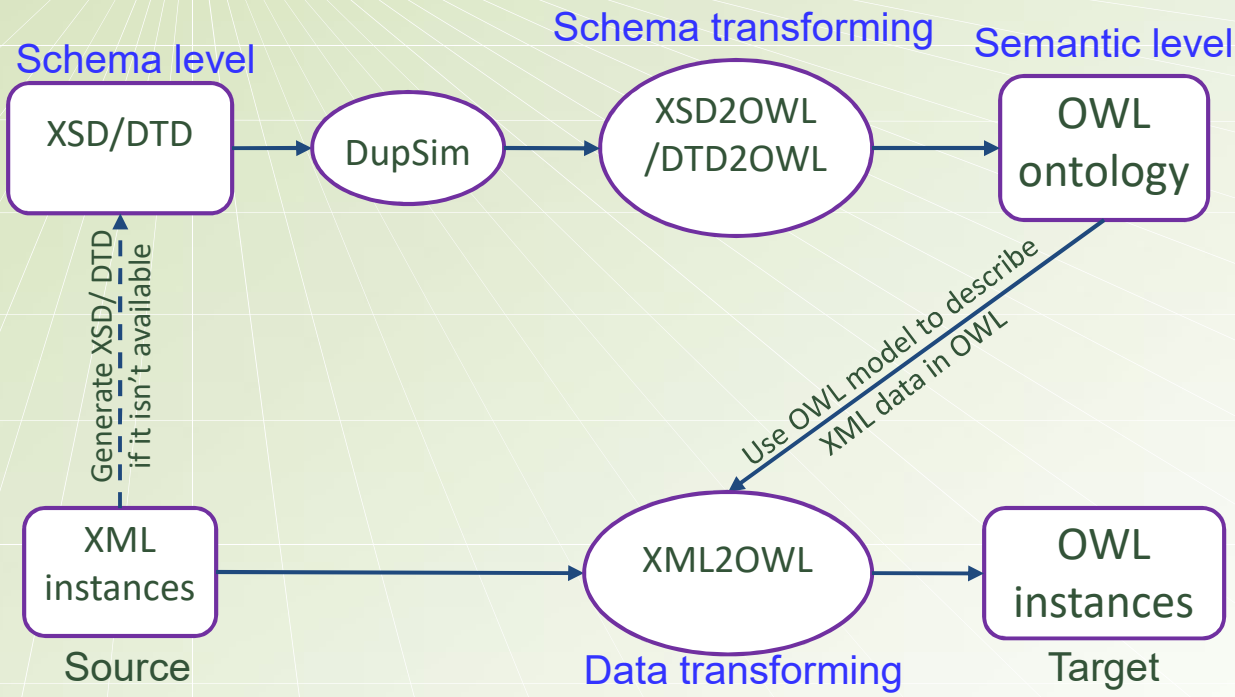
$$ChSim(e_1, e_2) = \frac{sum\_links(e_1, e_2) + sum\_links(e_2, e_1))}{leaves(e_1) + leaves(e_2)} \quad (30)$$

where *leaves(e_1)* is the total number of leaves in the sub-tree rooted at element e_1; *sum_links(e1, e2)* is the total number of links from the leaves of element e_1 to the leaves of element e_2.

S-Trans: Duplicate Similarity Measure and XML Transformation into OWL Ontology

Pham Thi Thu Thuy, Young-Koo Lee, and Sungyoung Lee, "S-Trans: Semantic Transformation of XML Healthcare Data into OWL Ontology", Knowledge-Based Systems Journal (SCI, IF: 1.574), ISSN: 0950-7051, 2012.

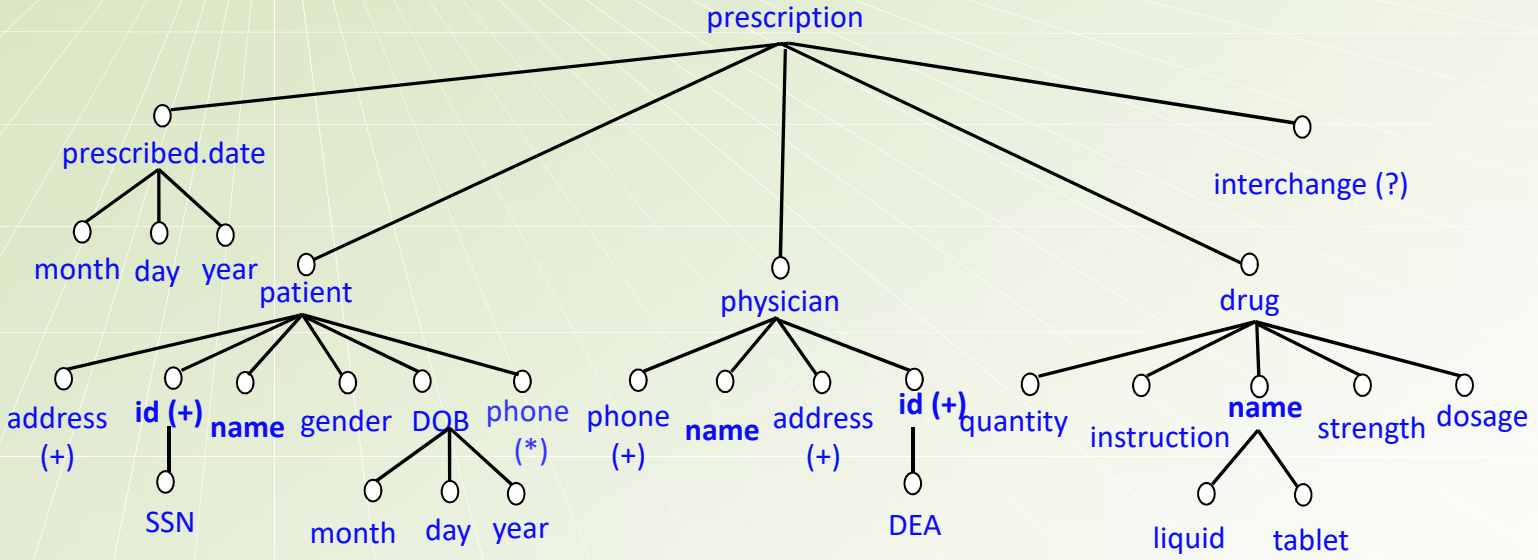
S-Trans framework



❖ Four steps: ① Extract schema; ② Measuring the duplicate similarity; ③ Transforming XSD to OWL model; ④Transforming XML instances into OWL individuals.

S-Trans: Motivating Example

- Determine the similarity among *address* elements of *patient* and *physician*., *prescription.dtd* [38]:



- Traditional approaches: address of physician is renamed as physician_address.
- But, two *address* elements represent the same information. If they are separated ➔ data redundancy.
- Ideal of XML transformation:
  - ✓ Correct
  - ✓ Complete
  - ✓ Unique representation for every object.

Duplicate Similarity

- Similar to our proposed ESim measure, the duplicate similarity (*DupSim*) measure is defined as the weighted sum of their semantic similarity (*SeSim*) and structure similarity (*StSim*):

$$DupSim(e_1, e_2) = \phi * SeSim(e_1, e_2) + (1 - \phi) * StSim(e_1, e_2) \quad (25)$$

where ϕ *is* the weight parameter.

- Difference from ESim, S-Trans only measures similarity of duplicates and within an XSD document $\Rightarrow$ *StSim* in S-Trans is different from ESim.

Structural Similarity

- Measure the similarity of elements having functional connected with current elements.
- Structure similarity *(StSim)* between two element e_1 and e_2 is defined as weighted sum of ancestor similarity *(AcSim)*, sibling similarity (SbSim), and children similarity (*ChSim*):

$$StSim(e_1, e_2) = \frac{\varepsilon_1 * AcSim(e_1, e_2) + \varepsilon_2 * SbSim(e_1, e_2) + \varepsilon_3 * ChSim(e_1, e_2)}{\varepsilon_1 + \varepsilon_2 + \varepsilon_3} \quad (12)$$

- The *AcSim*, *SbSim*, and *ChSim* are computed by collecting all their respective elements and then compared their semantic similarity.

Ancestor similarity algorithm

Purpose: Find the nearest common ancestor element

Input: *Two elements with the same name but in XSD tree,* e_1 and e_2

Output: *The ancestor similarity*

level== 0;max_level==11;

Function *ASim*(e_1, e_2, *level*)

if ((*SeSim*(e_1, e_2)==*1*)) or (*SeSim*(*parent::*e_1,*parent::*e_2)==*1*)

then return 1;

else if ((*SeSim*(e_1,*parent::*e_2)==*1*) **or** (*SeSim*(*parent::*e_1,*parent::*e_2)==*1*)

then return 0.85;

else if (*level == max_level*) **return** *0*;

else

return

power(0.85*****ASim*(parent::e_1, parent::e_2,level+1));

end;

End;

- Use post-order traversal to find the common ancestor.
- Compare the semantic similarity of each ancestor pair by using *SeSim* metric.

Sibling Similarity

- Assume that D_1 and D_2 are the collections of sibling elements of duplicates d_1 and d_2, respectively:

$$SbSim(D_1, D_2) = \begin{bmatrix} SeSim(e_{1_1}, e_{2_1}) & \cdots & SeSim(e_{1_1}, e_{2_n}) \\ \vdots & \ddots & \vdots \\ SeSim(e_{1_m}, e_{2_1}) & \cdots & SeSim(e_{1_m}, e_{2_n}) \end{bmatrix}, m \geq n \quad (26)$$

$$SbSim(D_2, D_1) = \begin{bmatrix} SeSim(e_{2_1}, e_{1_1}) & \cdots & SeSim(e_{2_1}, e_{1_m}) \\ \vdots & \ddots & \vdots \\ SeSim(e_{2_n}, e_{1_1}) & \cdots & SeSim(e_{2_n}, e_{1_m}) \end{bmatrix}, m < n \quad (27)$$

where *m* and *n* are the total number of sibling elements of d_1 and d_2, respectively

$$SbSim(D_1, D_2) = \frac{\sum_{i=1}^{m} \max_{j=1}^{n} (SeSim(e_{1_i}, e_{2_j}))}{m}, m \geq n \quad (28)$$

$$SbSim(D_2, D_1) = \frac{\sum_{i=1}^{n} \max_{j=1}^{m} (SeSim(e_{2_i}, e_{1_j}))}{n}, m < n \quad (29)$$

Children Similarity

- Assume that E_1 and E_2 are the collections of children of e_1 and e_2, respectively:

$$ChSim(E_1,E_2)=\begin{bmatrix} SeSim(e_{1_1},e_{2_1}) & \cdots & SeSim(e_{1_1},e_{2_n}) \\ \vdots & \ddots & \vdots \\ SeSim(e_{1_m},e_{2_1}) & \cdots & SeSim(e_{1_m},e_{2_n}) \end{bmatrix}, m \geq n \quad (21)$$

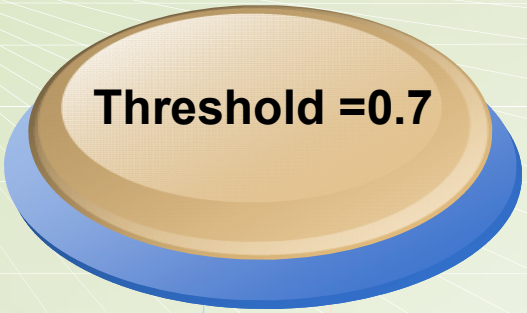
$$ChSim(E_2,E_1)=\begin{bmatrix} SeSim(e_{2_1},e_{1_1}) & \cdots & SeSim(e_{2_1},e_{1_m}) \\ \vdots & \ddots & \vdots \\ SeSim(e_{2_n},e_{1_1}) & \cdots & SeSim(e_{2_n},e_{1_m}) \end{bmatrix}, m < n \quad (22)$$

where *m* and *n* are the total number of children elements of e_1 and e_2, respectively

$$ChSim(E_1,E_2)=\frac{\sum_{i=1}^{m}\max_{j=1}^{n}(SeSim(e_{1_i},e_{2_j}))}{m}, m \geq n \quad (23)$$

$$ChSim(E_2,E_1)=\frac{\sum_{i=1}^{n}\max_{j=1}^{m}(SeSim(e_{2_i},e_{1_j}))}{n}, m < n \quad (24)$$

Similarity Classification



≥ 0.7: similarity:
procedure uses *owl:unionOf* to connect the parent nodes of these duplicates in the same domain.

< 0.7: non-similarity
renames the duplicated element by adding the parent element's name along with an underscore '_' character between the parent's name and the duplicate's name

```
<owl:DatatypeProperty rdf:about="#name">
   <rdfs:domain>  <owl:Class>
          <owl:unionOf rdf:parseType="Collection">
              <owl:Class rdf:about="#physician"/>
              <owl:Class rdf:about="#patient"/>
          </owl:unionOf>  </owl:Class>
   </rdfs:domain>
   <rdfs:range rdf:resource=
                  "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:ID="drug_name">
    <rdfs:range rdf:resource=
                  "http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#drug"/>
</owl:DatatypeProperty>
```

XSD2OWL Transforming Models

DTD (XML Document Type Definition)		XSD	OWL representation		
			Type	rdfs:domain	rdfs:range
DOCTYPE (root)		element@name, complexType	owl:class, owl:disjointWith owl:ObjectProperty	class name	child name
ELEMENT contains	other elements	element@name, complexType	owl:class, owl:disjointWith owl:ObjectProperty	class name	child name
	ENTITY reference only	element@ref	owl:class, owl:disjointWith		
	data type only	element@name,complexType mix="true"| simpleType|element@type	owl:DatatypeProperty	attribute name	datatype
ENTITY contains	>1 attributes	element@name,complexType mix="true" simpleType| >1 attribute@name	owl:DatatypeProperty, owl:subPropertyOf	attribute name	datatype
	one attribute	element@name,complexType mix="true"| simpleType|element@type	owl:DatatypeProperty	attribute name	datatype
ATTLIST contains	other property	>1 attribute@name, extension@base| restriction@base	owl:DatatypeProperty rdfs:subPropertyOf	attribute name	datatype
	data type only	1 attribute name	owl:DatatypeProperty	attribute name	datatype
	ENTITY reference	attribute@ref	owl:class owl:ObjectProperty	class name	child name
ELEMENT element-name (child1,child2,...)		sequence	owl:intersectionOf		
		choice	owl:unionOf		
+, *, ?		maxOccurs| minOccurs	owl:maxCardinality owl:minCardinality		

Experiments

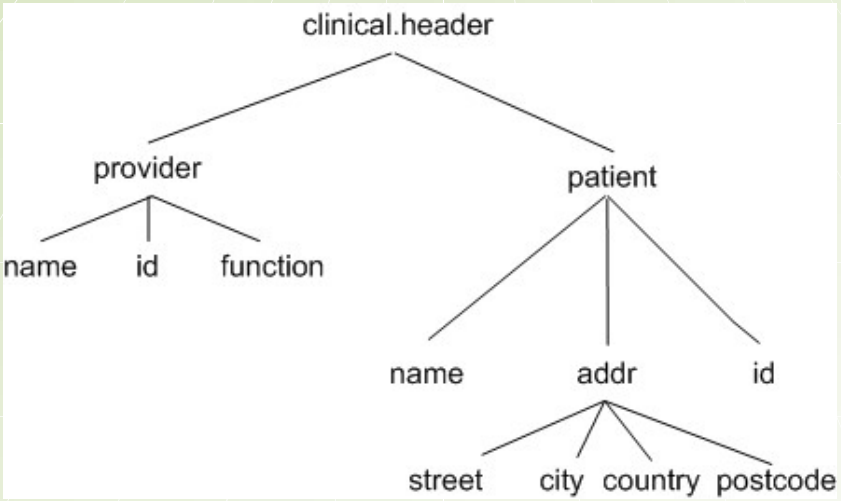
	Experimental setup	Determine parameter	Evaluation
ESim	- Synthetic XSDs. - Mutual similarity - Real-world XSDs	Weight parameters	- Compare with related work - Evaluate single factor
S-Trans	- Real-world XSDs/DTDs - Mutual duplicate similarities.	Classifying value	- Compare with related work - Evaluate single factor

Experimental Setup

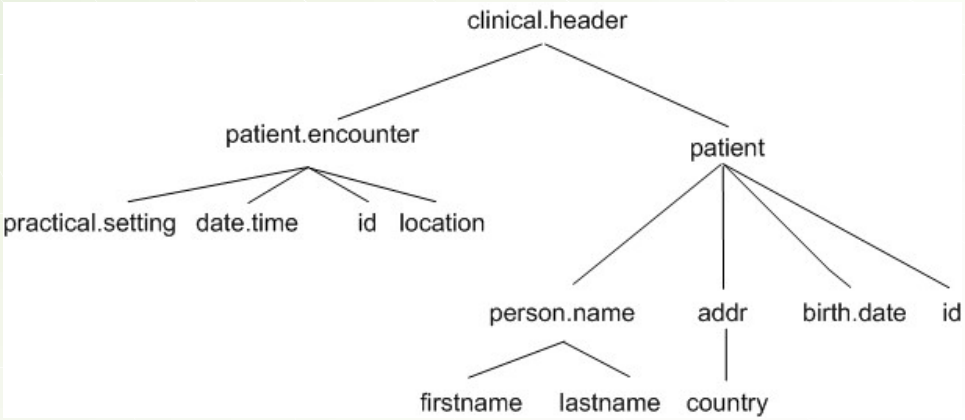
- Language: C# and XSLT (eXtensible Stylesheet Language Transformation).
- To measure the semantic similarity of element names, we integrated WordNet [18] and its .NET API by Simpson & Crowe [41]
- Dataset: XSD and XML documents downloaded from [38-40]
- We evaluate ESim and S-Trans by matching two XSD documents and matching between XSD and OWL ontology, respectively.

ESim: Determine weight values

- Determine values of parameters:
  - α and second parameter in ESim equation (1)
  - β, χ, and third parameter in SeSim equation (2)
- Prepare a synthesis XSDs (patients A, B, C, and D). Patient_A and Patient_B are presented in slide #13.



Tree representation for Schema Patient_C



Tree representation for Schema Patient_D

Determine weight values (cont')

- The mutual user-specified similarities for 6 pairs of schemas are listed in following table:

	Patient_A	Patient_B	Patient_C	Patient_D
Patient_A	1	0.86	0.48	0.5
Patient_B	0.86	1	0.4	0.37
Patient_C	0.48	0.4	1	0.8
Patient_D	0.5	0.37	0.8	1

Determine weight values (cont')

- The most reasonable values of α corresponding to similarity results expected by a user are represented using the black dots and occur within the interval of [0.5, 0.65]. Therefore, we choose $\alpha = 0.55$

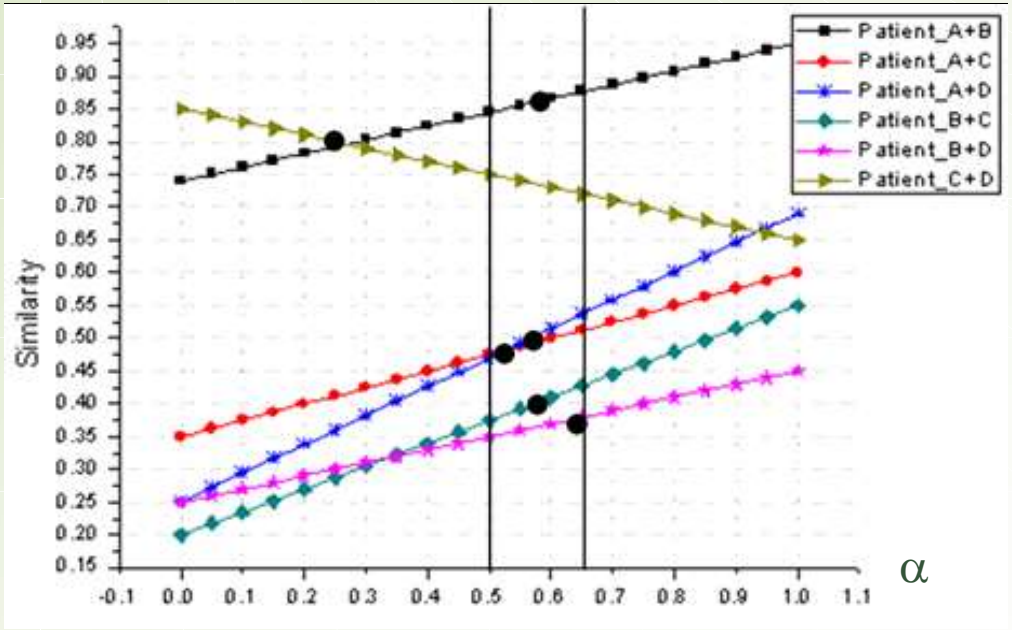


Figure. Determining weights of ESim function

Determine weight values (cont')

- The most reasonable values of β corresponding to similarity results expected by a user are represented using the black dots and occur within the interval of [0.3, 0.45]. Therefore, we choose $\beta = 0.4. \Rightarrow \chi = 0.3$

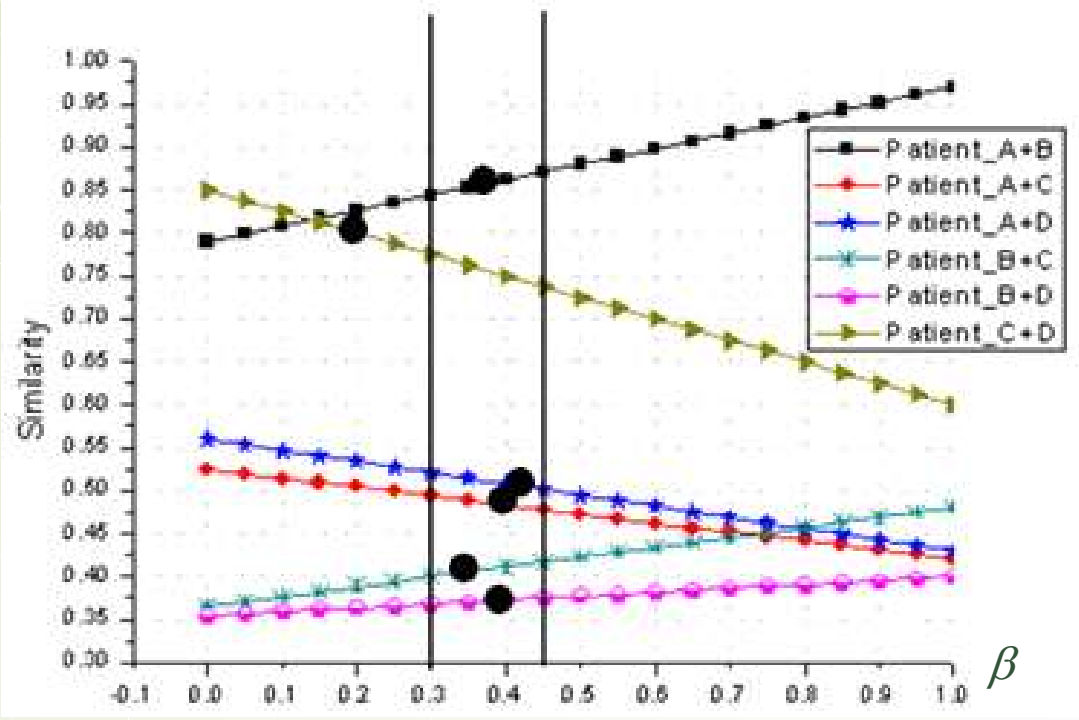


Figure. Determining weights of SeSim function

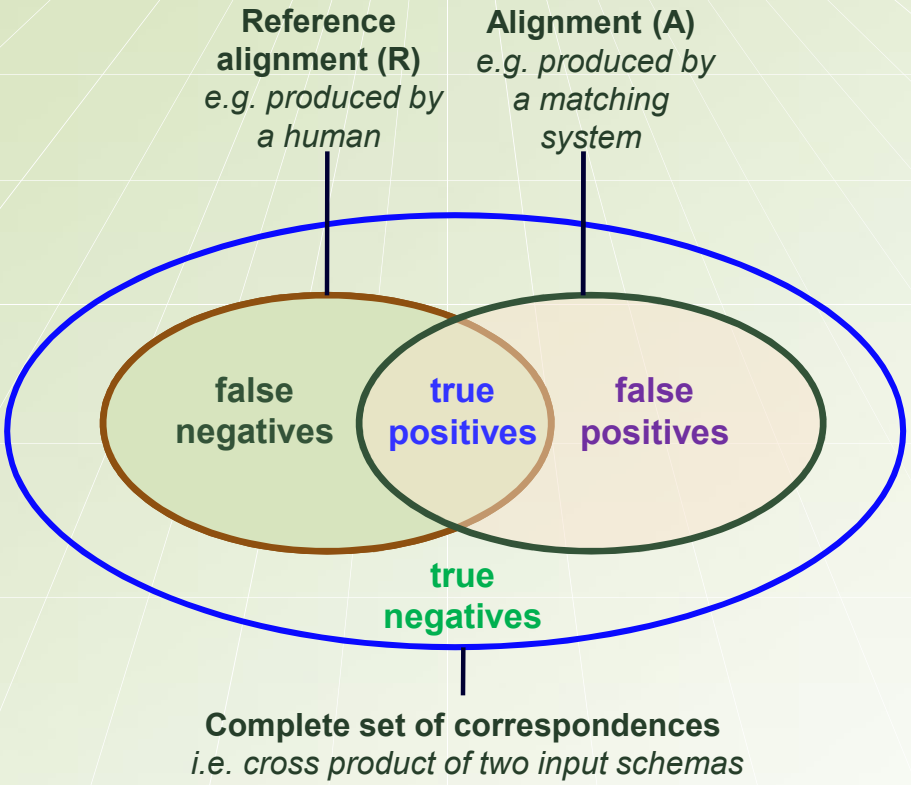
ESim: Real Dataset

- The mutual user-specified similarities for 6 pairs of schemas are listed in following table:

	Schema 1 vs Schema 2	#nodes	Average #nodes	#max depth
1	Patient A vs Patient B	12/10	11	3/3
2	Healthcaremetadata vs healthcarevocabulary	137/29	83	7/4
3	Yahoo Finance vs Standard	10/16	13	2/2
4	Cornell vs Washington	34/39	36	3/3
5	CIDX vs Excel	30/40	35	3/4
6	Google vs Looksmart	706/1081	893	11/16
7	Google vs Yahoo	561/665	613	11/11
8	Yahoo vs Looksmart	74/140	107	8/10
9	Iconclass vs Aria	999/553	776	9/3

Evaluation measures

- Evaluate similarity measure by matching two XSDs.
- Assess the quality of matching system by precision, recall, F-measure, and overall [32]



$$precision(A,R) = \frac{|R \cap A|}{|A|} \quad (31)$$

$$recall(A,R) = \frac{|R \cap A|}{|R|} \quad (32)$$

$$F-measure = 2 * \frac{precision * recall}{precision + recall} \quad (33)$$

$$overall = recall * \left(2 - \left(\frac{1}{precision}\right)\right) \quad (34)$$

Evaluation measures (cont')

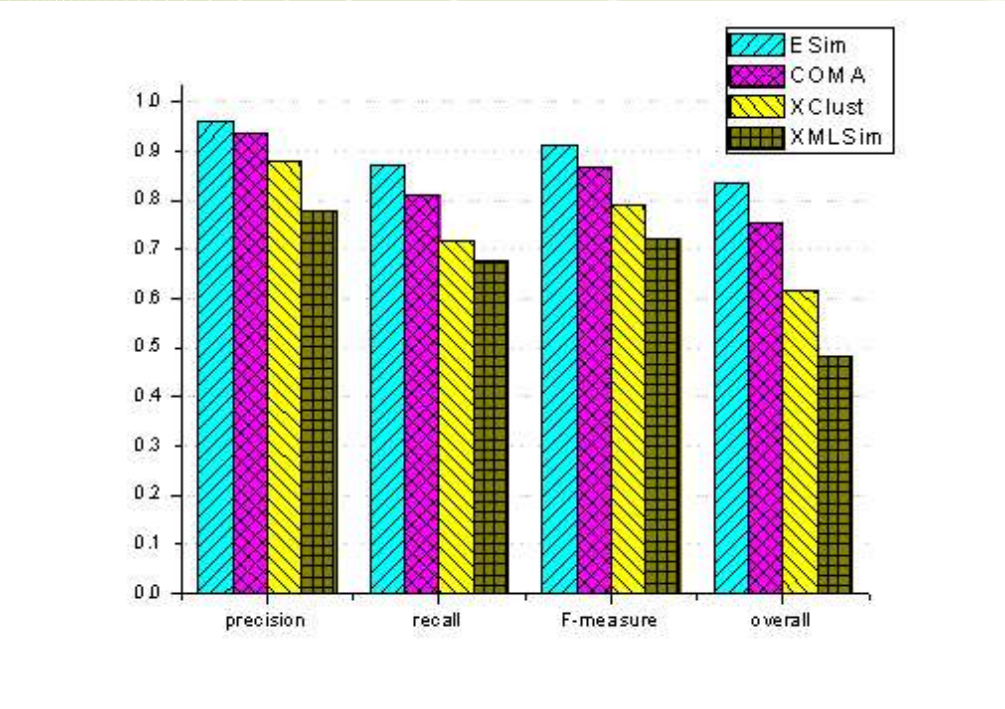
- Since 9 pairs of XSDs are used, precision and recall are the average value of those pairs.
- Propose weighted average equations:

$$precision = \frac{\sum_{i=1}^{n}(W_i * precision_i)}{\left|\sum_{i=1}^{n} W_i\right|} \quad (35)$$

$$recall = \frac{\sum_{i=1}^{n}(W_i * recall_i)}{\sum_{i=1}^{n} W_i} \quad (36)$$

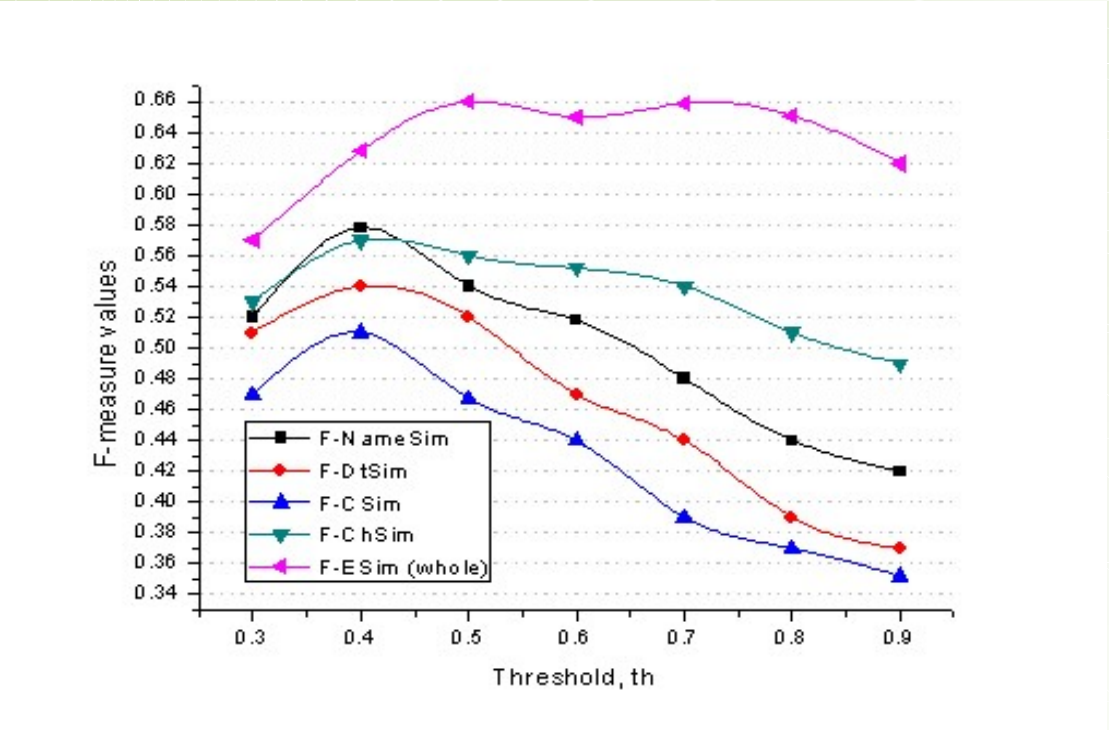
where n is # test cases (n=9); W_i is # correct matches of schema pair #i; $precision_i$ & $recall_i$ are precision & recall scores of schema pair #i

ESim: Compare with related work



- Matching comparisons of ESim to COMA [32], XMLSim [8], and XClust [7].
- XClust: structure measure, no Data type & semantic of elements measures
- XMLSim: only semantic measure of element name.

ESim: Quality of single measure



- F-ESim (combination of all factors): highest F-measure value
- Among factors, *F-measure* of children measure (ChSim) produces greatest values.

S-Trans: Experiments

S-Trans Dataset

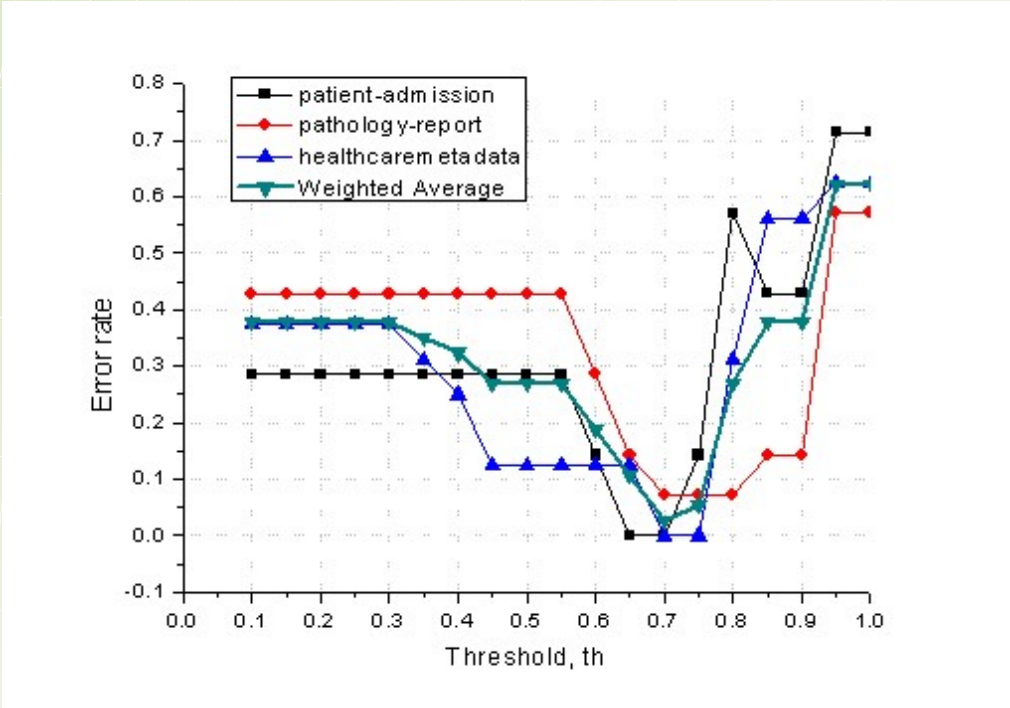
Table. Characteristics of the tested schemas [38-40]

	Schema name	File size (KB)	# nodes	max depth	# duplicates
1	drug_medicament	180	683	90	0
2	Patient-admission	40	240	4	7
3	healthcaremetadata	5523	137	7	16
4	pathology.report	328	778	5	14

Evaluate the proposed transforming strategies by matching an XSD document with an OWL ontology to determine the true matches, and compare our results with related methods.

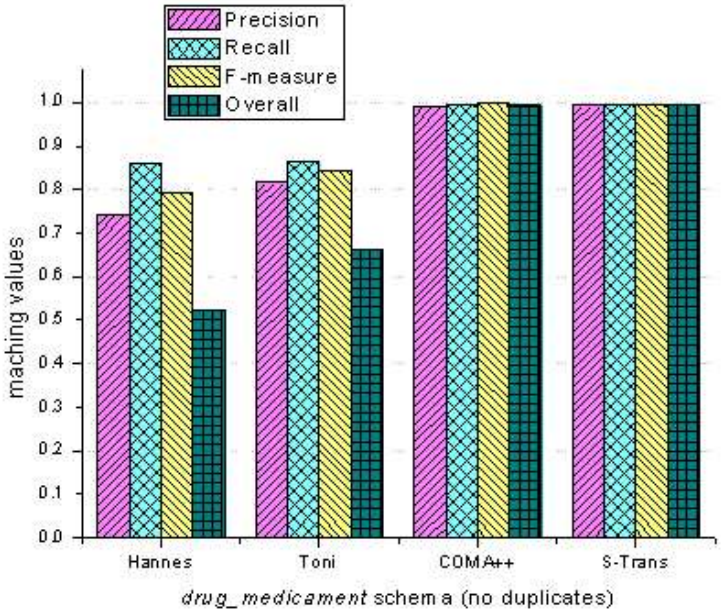
Determine classification value

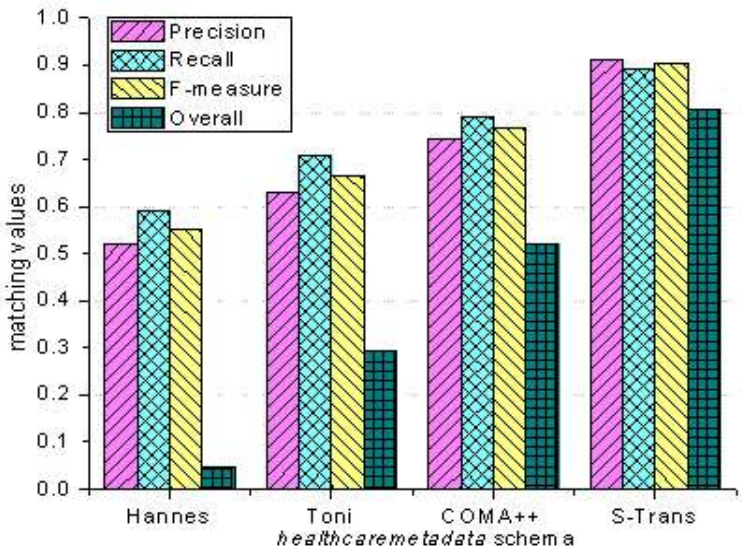
- First, manually classify duplicates into two groups: similar and non-similar.
- Second, compute classification error rate at each threshold, ranging between 0.1, 0.15, 0.2, …, 1.0



- Since error rate of classification achieves the minimum value at the threshold of 0.7, we use 0.7 as the classifying value to separate the duplicates into two groups, similar and non-similar.

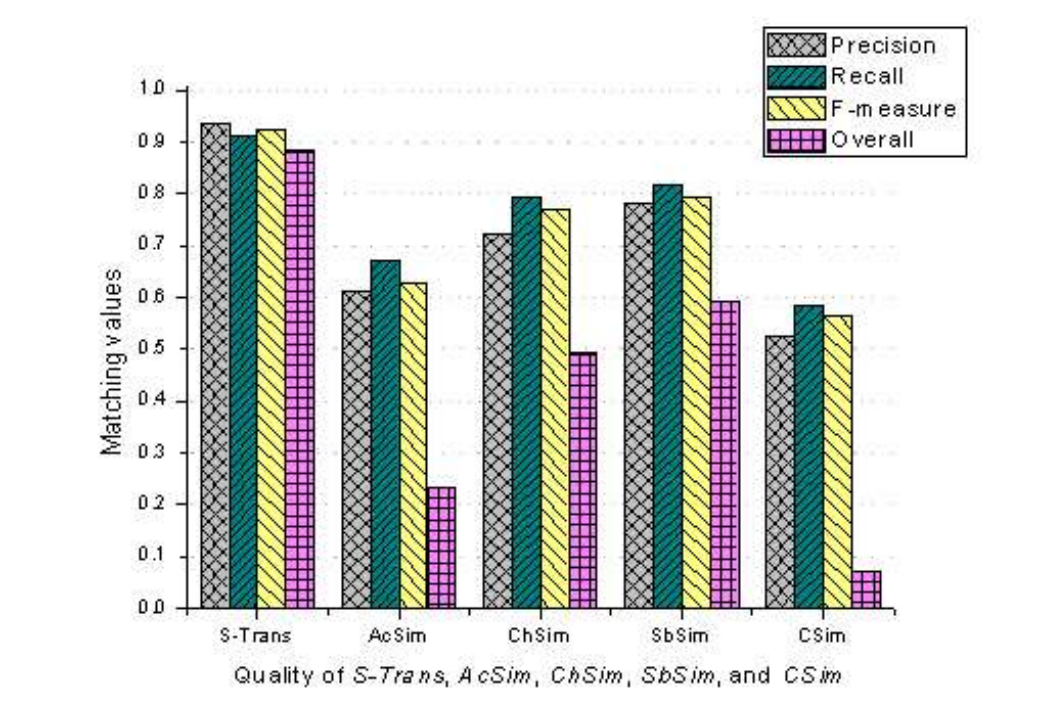
S-Trans & related work





- Compare S-Trans with one transforming method proposed by Hannes et al. [9] and with two matching methods introduced by Toni et al. [12] and COMA++ [29]
- No duplicates: S-Trans ≅ COMA++

S-Trans: Quality of single measure



- Among individual factor, sibling similarity (*SbSim*) gives highest quality:
  - Most duplicates have similar children and constraint
  - Small number of ancestor $\Rightarrow$ less influence to similarity values
- S-Trans: combination give highest matching quality.

Conclusions

- Propose two effective methods to enhance XML data sharing: ESim and S-Trans.
- ESim:
  - Propose a complete hybrid similarity measure framework.
  - Propose method to balance similarity factors.
  - Provide novel metrics to compute the Data type and cardinality constraint similarities.
  - Conduct a set of experiments to compare ESim with the related works and determine important role of each measuring factor.

Conclusions (cont')

- S-Trans:
  - Discover the similarity of XML duplicates.
  - Resolve duplicate problem: propose novel metrics to measure the semantic similarity between each duplicate pair.
  - Determine the classification value (0.7)
  - Experiments reveal:
    - Quite similar with human judgment
    - Overcome related methods
    - Determine the important measuring factor (Sibling measure, SbSim)

Future work

- Measuring the similarity of different data models (eg. DB vs XML, XML vs OWL)
- Matching of different data models: Relying on similarity score results to find the matches between source and target.
- Measuring the similarity of Web pages: Extending the semantic & structural similarity measures to compute the resemblance of textual content, structural layout, and query terms contained within pages.

References

1. T. Dalamagas, T. Cheng, K.-J. Winkel, T. Sellis, “A methodology for clustering XML documents by structure”, Inform. Syst. 31(3) 2006 187-228.
2. M.L. Lee, L. H. Yang, W. Hsu, X. Yang, “XCLust: Clustering XML Schemas for Effective Integration”, ACM, pp. 292—299, 2002.
3. Joe Tekli, Richard Chbeir, and Kokou Yetongnon, “A Hydrid Approach for XML Similarity”, SOFSEM '07 Proc. of the 33rd conference on Current Trends in Theory and Practice of Computer Science, Springer-Verlag Berlin, pp.783~795, 2007.
4. Joe Tekli, Richard Chbeir, and Kokou Yetongnon, “An overview on XML similarity: background, current trends and future directions”, Computer Science Review 3, pp. 151-173, 2009.
5. A. Fernandez, A. Polleres, and M. Blettner, “Towards Fine-grained Service Matchmaking by Using Concept Similarity”, Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web, pp. 31-45, 2007.
6. I. Choi, B. Moon, H.-J. Kim, “A clustering method based on path similarities of XML data”, Data Knowl. Eng. 60 (2) (2007) 361-376.
7. Mong Li Lee, Liang Huai Yang, Wynne Hsu, Xia Yang, “XCLust: Clustering XML Schemas for Effective Integration”, ACM Press, pp. 292—299, 2002.
8. Dongqiang D. Yang and David M.W. Powers, “Measuring Semantic Similarity in the Taxonomy of WordNet”, The 28th Australasian Computer Science Conference, pp. 315-322, 2005.
9. J. Madhavan, P. A. Bernstein, A. Doan, A. Halevy, “Corpus-based schema matching”, ICDE’05: Proceedings of the 21st Int. Conf. on Data Engineering, IEEE CS, USA, pp. 57-68, 2005.
10. S. Yi, B. Huang, W. T. Chan, “XML application schema matching using similarity measure and relaxation labeling”, Inform. Sci. 169 (1-2) (2005) 27-46.
11. B. Jeong, D. Lee, H. Cho, J. Lee, “A novel method for measuring semantic similarity for XML schema matching”, Expert Syst. Appl. 34 (3) (2008) 1651-1658.
12. R. Nayak, W. Iryadi, “XML schema clustering with semantic and hierarchical similarity measure”, Knowledge-Based Syst. 20 (2007) 336-349.
13. A. Wojnar, I. Mlynkova, J. Dokulil, “Similarity of DTDs based on edit distance and semantics”, IDC’08, Studies in Computation Intelligence, vol. 162, Springer-Verlag, Catania, Italy, pp. 207-216, 2008.
14. A. Wojnar, I. Mlynkova, J. Dokulil, “Structural and semantic aspects of similarity of document type definition and XML schema”, Inform. Sci. 180 (2010) 1817-1836.
15. A. Algergawy, R. Nayak, G. Saake, “Element similarity measures in XML schema matching”, Inform. Sci., pp. 4975-4998, 2010.
16. T.-S. Kim, J.-H. Lee, J.-W. Song, “Semantic structural similarity for clustering XML documents”, Int. Conf. on Convergence and Hybrid Inf. Tech., 2008 IEEE.
17. L. Song, J. Ma, J. Lei, D. Zhang, Z. Wang, “Semantic structural similarity measure for clustering XML documents”, WISM 2009, LNCS 5854, pp. 232-241.
18. Princeton University, WordNet_ A lexical database for English, http://wordnet.princeton.edu/wordnet, visited 12 May 2011.
19. M. Ferdinand, C. Zirpins, D. Trastour, “Lifting XML Schema to OWL”, Proceedings of 4th ICWE, pp. 354-358, 2004.
20. Hannes Bohring, S¨oren Auer, “Mapping XML to OWL Ontologies”, Marktplatz Internet: Von e-Learning bis e-Payment, Leipziger Informatik-Tage, Germany, 147-156, 2005.

References (cont')

21. C. Tsinaraki, S. Christodoulakis, "XS2OWL: A Formal Model and a System for Enabling XML Schema Applications to Interoperate with OWL-DL Domain Knowledge and Semantic Web Tools", Proceedings of DELOS, 137-146, 2007.
22. Hannes Bohring, S¨oren Auer, "Mapping XML to OWL Ontologies", Marktplatz Internet: Von e-Learning bis e-Payment, Leipziger Informatik-Tage, Germany, 147-156, 2005.
23. C. Tsinaraki, S. Christodoulakis, "XS2OWL: A Formal Model and a System for Enabling XML Schema Applications to Interoperate with OWL-DL Domain Knowledge and Semantic Web Tools", Proceedings of DELOS, 137-146, 2007.
24. Bernd A., Catriel B., Irini F., Michel S., "Ontology-Based Integration of XML Web Resources", The First International Semantic Web Conference, Springer-Verlag, 117-131, 2002.
25. Toni Rodrigues, P. Rosa, and J. Cardoso, "Moving from syntactic to semantic organizations using JXML2OWL", Journal of Computers in Industry 59, 808-819, 2008.
26. C. Cruz, C. Nicolle, "Ontology Enrichment and Automatic Population from XML Data", 4th International VLDB Workshop on Ontology-based Techniques for Databases in Information Systems and Knowledge Systems, pp. 17-20, 2008.
27. P. T. T. Thuy, Y-K Lee and S.Y Lee, "DTD2OWL: Automatic Transforming XML Documents into OWL Ontology", International Conference on Interaction Science, ACM, 125-131, 2009
28. R. Rada, H. Mili, E. Bicknell, and M. Blettner "Development and application of a metric on semantic nets", IEEE Transactions on Systems, Man and Cybernetics, 19(1):17–30, 1989.
29. Z. Wu and M. Palmer, "Verbs semantics and lexical selection", Proceedings of 32nd Computational Linguistics, pp.133–138, 1994.
30. C. Leacock and M. Chodorow, "Combining local context with WordNet similarity for word sense identification", MIT Press, pp. 305-332, 1998.
31. Y. Li, Z. Bandar, and D. McLean, "An approach for measuring semantic similarity between words using multiple information sources", IEEE Trans. Knowl. Data Eng., 15(4):871–882, 2003 .
32. H-H Do, and E. Rahm, "COMA - A System for Flexible Combination of Schema Matching Approaches", VLDB, pp 610–621, 2002.
33. Database group Leipzig, "COMA++", http://dbs.uni-leipzig.de/Research/coma.html
34. D. D. Yang and D. M.W. Powers, "Measuring Semantic Similarity in the Taxonomy of WordNet", ACSC2005, pp. 315-322, 2005.
35. Richi Nayak, Wina Iryadi, "XML Schema clustering with semantic and hierarchical similarity measure", Knowledg-Based Systems 20 (2007) 336-349.
36. D Vint Productions, XML Schema - Data Types Quick Reference, http://www.xml.dvint.com (2003).
37. A BackOffice Associates, LLC Company, "HIT software", http://www/hitsw.com/xml_utilities
38. Robin Cover, "The Cover Pages: Schema for Patient Medical Record", http://xml.coverpages.org/BordenASTM20010314.html
39. OSOR Forge Hospital, "SCM Repository", http://forge.osor.eu/plugins/scmsvn/viewcvs.php
40. Mebiquitous XML Schema, http://ns.medbiq.org
41. T. Simpson, M. Crowe, "WordNet.Net", 2005, http://opensource.ebswift.com/WordNet.Net
42. Database group Leizig, "COMA++", http://dbs.uni-leizig.de/Research/coma.html
43. MSDN Microsoft, "maxOccurrs Property", http://msdn.microsoft.com/en-us/library/windows/desktop/ms759115%28v=vs.85%29.aspx

Publications

Journals

1. Pham Thi Thu Thuy, Young-Koo Lee, and Sungyoung Lee, "Semantic and Structural Similarity Analysis for Integration of Ubiquitous XML Healthcare Data", Journal of Personal and Ubiquitous Computing (SCIE, IF: 1.137), ISSN: 1617-4909, 2012. *(accepted)*
2. Pham Thi Thu Thuy, Young-Koo Lee, and Sungyoung Lee, “S-Trans: Semantic Transformation of XML Healthcare Data into OWL Ontology”, Knowledge-Based Systems Journal (SCI, IF: 1.574), ISSN: 0950-7051, 2012. http://dx.doi.org/10.1016/j.knosys.2012.04.009

Conferences

1. Pham Thi Thu Thuy, Young-Koo Lee and Sungyoung Lee, "DTD2OWL: Automatic Transforming XML Documents into OWL Ontology", ICIS 2009, ACM, pp. 125-131, Seoul, Korea, November 24-26, 2009.
2. Pham Thi Thu Thuy, Young-Koo Lee, and Sungyoung Lee, "XSD2RDFS and XML2RDF Transformation: a Semantic Approach",Proceedings of the 2nd International Conference on Emerging Databases (EDB 2010),pp. 167-172, Jeju, Korea, August 30-31, 2010.
3. Pham Thi Thu Thuy, Young-Koo Lee, and Sungyoung Lee, "Semantic and Structural Similarities between XML Schemas for Integration of Ubiquitous Healthcare Data", The 2011 FTRA Internatioal Workshop on U-Healthcare Technologies and Services (U-Healthcare 2011), Jeju, Korea, December 12-15, 2011
4. Pham Thi Thu Thuy, Young-Koo Lee, and Sungyoung Lee, "R2Sim: A Novel Semantic Similarity Measure for Matching between RDF Schemas", The 2012 FTRA International Conference on Advanced IT, engineering and Management (FTRA AIM 2012), Seoul, Korea, February 6-8, 2012.
5. Pham Thi Thu Thuy, Young-Koo Lee and Sungyoung Lee, " Extracting OWL Ontology from XML instances via XML Schema", The 32nd KIPS and Fall Conference, Seoul, Korea, pp. 801-802, November 2009.

Thank You !

감사합니다 !

Revisions made for public presentation

Professor Byeong-soo Jeong's comments:

- How to determine values for weight factors?

Answer:

- Most of the current paper claim that the setting of similarity parameters can be determined by user and, hence, it is not discussed.
- The problem is how to prepare a reasonable setting so that the similarity measure returns reasonable results.
- For this purpose, we use the following strategy: Firstly, we prepare a set of synthetic XSDs and we determine their mutual similarity from user's perspective. Then, we set the respective parameters so that the similarity measure returns similar results.

Revisions made for public presentation

Professor Choong-Seon Hong's comments:

- Review more papers on semantic similarity measures.

Answer:

- More related papers have been added to the presentation and the thesis.
  - ✓ Distance based semantic [13, 14]
  - ✓ Cosine patch matching [16, 17]
  - ✓ Semantic based [12, 15]