XML Schema Element Similarity Measures: A Schema Matching Context

Alsayed Algergawy[1], Richi Nayak[2], and Gunter Saake[1]

[1] Otto-von-Guericke University, 39106 Magdeburg, Germany
[2] Queensland University of Technology, 2434 Brisbane, Australia
{alshahat,saake@iti.cs.uni-magdeburg.de}, r.nayak@qut.edu.au

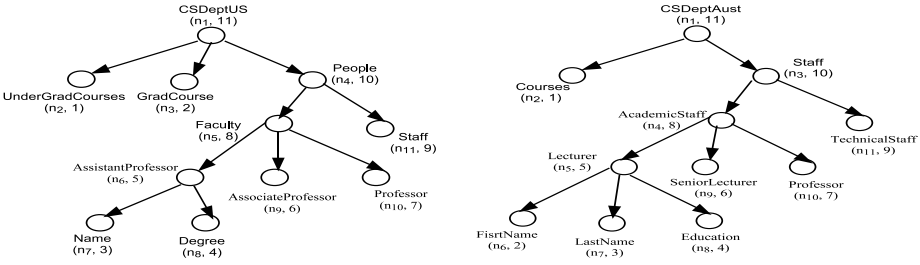
Abstract. In this paper, we classify, review, and experimentally compare major methods that are exploited in the definition, adoption, and utilization of element similarity measures in the context of XML schema matching. We aim at presenting a unified view which is useful when developing a new element similarity measure, when implementing an XML schema matching component, when using an XML schema matching system, and when comparing XML schema matching systems.

1 Introduction

Schema matching plays a central role in many data-shared applications such as, data integration, data warehousing, E-business, Semantic Web, data migration, and XML data clustering [14,8]. Due to the complexity inherent in schema matching, it is mostly performed manually by a human expert. However, manual reconciliation tends to be a slow and inefficient process especially in large-scale and dynamic environments. Therefore, the need for automating schema matching has become essential. Consequently, a myriad of matching algorithms have been proposed and many systems for automatic schema matching have been developed, such as Cupid [12], COMA/COMA++ [6], LSD [7], SMatch [10], and PORSCHE [15]. The common trait among these systems is that they all exploit schema element features (properties) as well as the relationships between schema elements utilizing different element similarity measures.

A few studies have been conducted to report and evaluate element similarity measures independent of their matching systems. Some of them [5,2] reported results comparing whole matching systems without considering individual element measures. The work proposed in [9] presents a library of element level semantic matchers implemented within the S-Match [10] system considering only the element features. Recently, there is a few work that survey approaches assessing the similarity between XML data [16]. However, this work focuses on measuring the similarity between whole XML data not on the individual elements. In this paper, we aim to classify, review, and experimentally compare element similarity measures in the context of XML schema matching. This study is guided by the following observation: a number of element similarity measures working on

R. Meersman, T. Dillon, P. Herrero (Eds.): OTM 2009, Part II, LNCS 5871, pp. 1246–1253, 2009.



(a) Schema Tree ST1 (b) Schema Tree ST2

Fig. 1. Tree representation of XML schemas

element features exploit only the internal element properties without considering its surrounds. While, the other element similarity measures exploit element relationships considering the element surrounds.

2 Preliminaries

An XML schema can be modeled as a graph. It can also be represented as a tree by dealing with nesting and repetition problems using a set of predefined transformation rules [11]. Consequently, in the following, we represent XML schemas as rooted, labeled trees, called schema trees, ST, defined as $ST = \{N_T, E_T, Lab_N\}$, where N_T is a set of nodes, E_T is a finite set of edges, each edge represents the relationship between two nodes, and Lab_N is a finite set of node labels. In this paper, we use the widely used two XML schemas that represent the organization in universities from different countries [7,1] to show the effect of various measures. We use the postorder traversal to uniquely number tree nodes. Figs. 1(a,b) show the schema trees of the two XML schemas, wherein each node is associated by its name label, such as $CSDeptUS$, its object identifier, such as n_1, and its corresponding postorder traversal number.

Given a schema tree of an XML schema, an *Element* ($\mathcal{E}\ell$) is a singular data item that is the basis of the similarity measures. The property set associated to each element is called the element *feature*. We categorize schema tree elements into: *atomic elements*, which represent simple elements or attribute nodes, and have no outgoing edges, and *complex elements*, which are the internal nodes in the schema tree. Furthermore, there exist many relationships among schema tree elements that reflect the hierarchical nature of the XML schema tree, such as *parent-child (induced)*, *ancestor-descendant (embedded)*, or *order* relationships.

To measure the similarity between schema tree elements, the element'features, and relationships among them should be exploited. A function, Sim, is a similarity measure that quantifies the similarity between elements. It is represented as $Sim(\mathcal{E}\ell_1, \mathcal{E}\ell_2)$, and its value is computed by the employed method. Usually, the similarity value ranges between 0 and 1, when the measure is normalized. The value of 0 means strong dissimilarity between elements, while the

value of 1 means exact same elements. The similarity between two elements $\mathcal{E}\ell_1 \in ST1, \mathcal{E}\ell_2 \in ST2$ can be determined using the following equation:

$$Sim(\mathcal{E}\ell_1, \mathcal{E}\ell_2) = w_I \times InterSim(\mathcal{E}\ell_1, \mathcal{E}\ell_2) + w_E \times ExterSim(\mathcal{E}\ell_1, \mathcal{E}\ell_2) \quad (1)$$

where $InterSim(\mathcal{E}\ell_1, \mathcal{E}\ell_2)$ represents the internal similarity measure between the two elements exploiting their features, while $ExterSim(\mathcal{E}\ell_1, \mathcal{E}\ell_2)$ represents the external similarity measure exploiting their hierarchal relationships, and w_I and w_E are weights to quantify the importance of each measure.

3 Internal Element Similarity Measures

The internal element measures exploit the element features, such as their *names*, *data types*, *constraints*, *annotations*, and others to compare elements from different schema trees. Depending on the type of exploited feature, we present the following internal measures.

3.1 Name Similarity Measure

Element names can be syntactically similar *(Staff, TechnicalStaff)* or semantically similar *(People, Staff)*. As a result, it is desirable to consider *syntactic* and *semantic* measures both to compute a degree of similarity between element names. In order to make element names comparable, they should be normalized into a set of tokens. After decomposing each element name into a set of tokens, the name similarity between the two sets of name tokens T_1 and T_2 is determined as the average best similarity of each token with all tokens in the other set [12,13]. It is computed as:

$$Nsim(T1, T2) = \frac{\sum_{t_1 \in T_1}[\max_{t_2 \in T_2} sim(t_1, t_2)] + \sum_{t_2 \in T_2}[\max_{t_1 \in T_1} sim(t_2, t_1)]}{|T1| + |T2|}$$

To determine the similarity between a pair of tokens, $sim(t_1, t_2)$, both syntactic and semantic measures can be used.

Syntactic measures (String-based). Syntactic measures take the advantage of the representation of element names as strings (sequence of characters). There are many methods to compare strings depending on the way the string is seen (as exact sequence of characters, an erroneous sequence of characters, and a set of characters), such as *Edit distance, Jaro similarity*, and *N-gram* [4,8,9,1].

Semantic measures (Language-based). The semantic measures are based on using Natural Language Processing (NLP) techniques to find the degree of similarity between schema tree element names. Most of these techniques heavily rely on the use of external sources, such as dictionaries and lexicons. Typically, WordNet is used either to simply find close relationships, such as synonym between element names, or to compute some kind of semantic distance between them. The SMatch system [10] proposes semantic schema matching that exploits

the features in WordNet as a background knowledge source to return semantic relations (e.g. equivalence, more general) between element names rather than similarity values in the [0,1] range. Another possibility is to utilize a domain-specific use-defined dictionary. COMA++ [6] and PORSCHE [15] utilize a user-defined dictionary to get a similarity degree between element names.

3.2 Data Type Similarity Measure

Although the element name is considered a necessary source for determining the element similarity, however, it is an insufficient source. For example, the name similarity between two elements $ST1.n_9$ and $ST2.n_3$, see Fig. 1, equals 1.0. This is a false positive match as these two elements are of different data types. This necessitates the need for other schema information sources used to prune some of these false positive matches. The element data type is another schema information source that makes a contribution in determining the element similarity. XML schema supports 44 primitive and derived built-in data types[1]. Using the XML built-in data type hierarchy, a data type similarity can be computed. One method is to build a data type similarity table similar to the used in [12,13] that includes the similarity between two data types.

3.3 Constraint Similarity Measure

Another schema information source of the element that makes another contribution in assessing the element similarity is its constraints. The cardinality (occurrence) constraint is considered the most significant. The *minOccurs* and *maxOccurs* in the XML schema define the minimum and maximum occurrence of an element that may appear in XML documents. A cardinality table for DTD constraints has been proposed in [11]. The authors of [13] adapt this table for the constraint similarity of XML schemas.

3.4 Annotation Similarity Measure

To enhance the internal element similarity, we capture the document information about schema elements existed in the annotation element. To this end, we make use of a token-based similarity measure. According to the comparison made in [4], the TFIDF ranking performed best among several token-based similarity measures. For this, we consider the TFIDF measure in our study.

4 External Element Similarity Measures

In contrast to internal element measures that exploit the element features without considering the position (context) of the element. The external measures make use of the element relationships instead of its features.

[1] `http://www.w3.org/TR/xmlschema-2/`

4.1 Element Context Measure

The context of an element is the combination of its *child*, *leaf*, *ancestor*, and *sibling* contexts. Two elements are structurally similar if they have similar contexts. To determine the context (structural) similarity between two elements $\mathcal{E}\ell_1 \in ST_1$ and $\mathcal{E}\ell_2 \in ST_2$, the similarity of their child, leaf, sibling, and ancestor contexts should be computed.

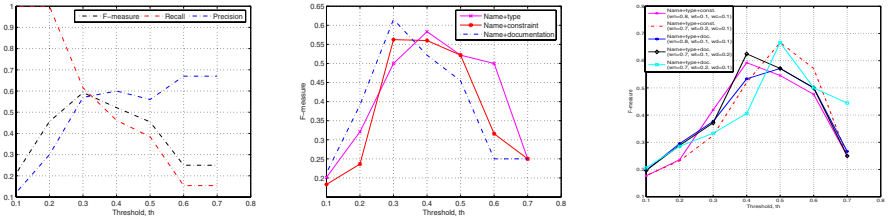
1. *Child context similarity.* The child context set (the set of its immediate children nodes including attributes and subelements) is first extracted for each element. The internal similarity between each pair of children in the two sets is determined, the matching pairs with maximum similarity values is selected, and finally the average of best similarity values is computed.
2. *Leaf context similarity.* First, the leaf context set (the set of leaf nodes of subtrees rooted at the element) is extracted for each element. Then, a suitable set comparison measure can be used. The authors in [1] convert the leaf context sets into numerical vectors and they apply the cosine measure.
3. *Sibling context similarity.* The sibling context set (contains both the preceding siblings and the following siblings) is extracted for each element is extracted. The internal similarity between each pair of siblings in the two sets is then determined, the matching pairs with maximum similarity values is selected, and finally the average of best similarity values is computed.
4. *Ancestor context similarity.* The ancestor context similarity captures the similarity between two elements based on their ancestor contexts. The ancestor context for a given element $\mathcal{E}\ell_i$ is the path extending from the root node to $\mathcal{E}\ell_i$. To compare between paths, authors in [3] established four scores, which then have been used in [1].

5 Experimental Evaluation

In order to evaluate the element similarity measures described in the paper, we carried out a set of experiments using the two schema trees shown in Fig. 1 and their characteristics represented in Fig. 2. The quality of the element similarity measures is verified using *F-measure*, a harmonic mean of *precision (P)* and *recall (R)*. The main objective of these evaluations is to extract several general rules that can be used as guides during the schema matching development.

	ST1				ST2			
OID	name	type	cardinality minOccurs	cardinality maxOccurs	name	type	cardinality minOccurs	cardinality maxOccurs
n_1	CSDeptUs	complex	0	unbounded	CSDeptAust	complex	0	unbounded
n_2	UnderGradCourses	string	0	1	Courses	string	0	1
n_3	GradCourse	string	0	1	Staff	complex	0	unbounded
n_4	People	complex	0	unbounded	AcademicStaff	complex	0	unbounded
n_5	Faculty	complex	0	unbounded	Lecturer	complex	0	unbounded
n_6	AssistantProfessor	complex	0	unbounded	FirstName	string	0	1
n_7	Name	string	0	1	LastName	string	0	1
n_8	Degree	string	0	1	Education	string	0	1
n_9	AssociateProfessor	string	0	1	SeniorLecturer	string	0	1
n_{10}	Professor	string	0	1	Professor	string	0	1
n_{11}	Staff	string	0	1	TechnicalStaff	string	0	1

Fig. 2. Schema tree characteristics



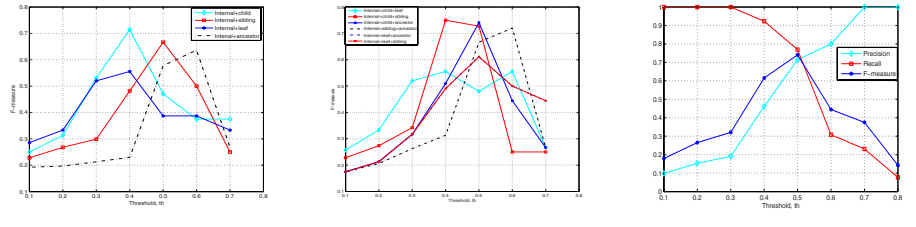
(a) Name measure (b) Name+ one of internal measures (c) Name+ two of internal measures

Fig. 3. Internal element similarity measures quality

5.1 Element Measures Quality

Internal measures without external information sources. The quality of each internal similarity measure (name, data type, documentation, and cardinality constraint) is first evaluated alone and then different combinations between them are also evaluated. The results of these evaluations are reported in Fig. 3. The name measure achieves F-measure ranging between 20% to 58% as shown in Fig. 3a, while the data type measure produces F-measures between 29% and 32%. To get better matching quality, different combinations have been used. First, the name measure is combined with one of the other internal measures. The results reported in Fig.3b show that the combined name and documentation measures performed better than the other two combinations. Then, the name measure is combined with two of the other measures. Fig. 3c illustrates that F-measure improves and its value reaches 67% when combining name, type, and documentation (constraint) measures. Using all internal measures improves F-measure to 72%.

Internal & external measures quality. The second set of experiments was implemented to observe the quality of internal element similarity measure with different combinations of external element measures. The results of these evaluations are reported in Fig. 4. Combining the leaf context with the internal



(a) Internal+one context measure (b) Internal+two context measures (c) Internal+ external measures

Fig. 4. Internal & external element similarity measures quality

measure deteriorates the matching quality, as shown in Fig. 4a, while the child context outperformed better than the other combinations. Fig. 4b shows that combining also the child context with another element context other than the leaf context surpasses the other combinations. Fig. 4c outlines the results produced by combining the internal and external measures. The figure presents an interesting finding regarding to the used threshold (th). Small values of threshold result in a large number of false positives (small precision values) and a small number of false negatives (large recall values). Increasing the value of threshold causes an opposite situation. The highest F-measure (0.76) was obtained at a threshold of 0.5.

Effect of external information sources. Although the used test schemas are small, matching is not of high quality due to different heterogeneities exist in the tested schemas. F-measure values range between 17% and 76% depending on the used element measures and the selected threshold. To improve the matching quality, one method is to use semantic measures. To this end, we built a domain-specific dictionary, and we developed another set of experiments to observe the effect of external information sources on matching quality. The results of these evaluation are reported in Fig. 5. Compared to results shown in Fig. 4, F-measure has nearly the same value with/without the external dictionary at a threshold value of 0.1. At higher threshold values, F-measure has been improved gradually. It increases from 26% to 30% at a threshold value of 0.2, from 61% to 65% at 0.4, and from 76% to 80% at 0.5. The best F-measure obtained is 80% at a threshold of 0.5 using the external dictionary, and 76% without the dictionary.

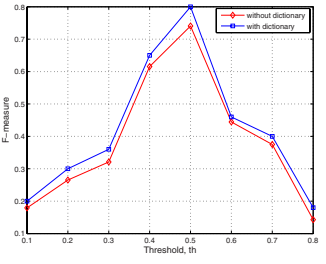


Fig. 5. Element similarity measures quality with an external dictionary

6 Discussion

Experiments we conducted present several interesting findings that can be used as a guide during schema matching development. These findings include: (1) Using a single element similarity measure is not sufficient to assess the similarity between XML schema elements. This necessitates the need to utilize several element measures exploiting both internal element features and external element relationships. (2) Utilizing several element measures provides the advantage of our matching algorithms to be more flexible. However, it also embeds a disadvantage of how to combine these similarity measures. In this study, We select the aggregation function (weighted-sum) as a combining strategy. Reported results demonstrate that the name measure has the most effect of the internal measure, while external measures are nearly of equal effect. (3) Selecting the candidate

correspondences is largely based on the value of *threshold*. Low values of threshold result in a large number of false positives (very low precision) and a small number of false negatives (high recall), while high values of threshold causes an inverse situation, as shown in Fig. 5. (4) Exploiting external information sources, such as WordNet or domain-specific dictionaries, improves the matching quality. However, to get this improvement, the matching efficiency do decline. In the large-scale context, a trade-off between matching effectiveness and matching efficiency should be considered.

References

1. Algergawy, A., Schallehn, E., Saake, G.: Improving XML schema matching using prufer sequences. DKE 68(8), 728–747 (2009)
2. Avesani, P., Giunchiglia, F., Yatskevich, M.: A large scale taxonomy mapping evaluation. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 67–81. Springer, Heidelberg (2005)
3. Carmel, D., Efraty, N., Landau, G.M., Maarek, Y.S., Mass, Y.: An extension of the vector space model for querying XML documents via XML fragments. SIGIR Forum 36(2) (2002)
4. Cohen, W.W., Ravikumar, P., Fienberg, S.E.: A comparison of string distance metrics for name-matching tasks. In: IIWeb, pp. 73–78 (2003)
5. Do, H.H., Melnik, S., Rahm, E.: Comparison of schema matching evaluations. In: the 2nd Int. Workshop on Web Databases (2002)
6. Do, H.H., Rahm, E.: Matching large schemas: Approaches and evaluation. Information Systems 32(6), 857–885 (2007)
7. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Ontology matching: A machine learning approach. In: Handbook on Ontologies, International Handbooks on Information Systems (2004)
8. Euzenat, J., et al.: State of the art on ontology alignment. In: Part of research project funded by the IST Program, Project number IST-2004-507482, Knowledge Web Consortim (2004)
9. Giunchiglia, F., Giunchiglia, F., Yatskevich, M., Yatskevich, M.: Element level semantic matching. In: ISWC workshops (2004)
10. Giunchiglia, F., Yatskevich, M., Shvaiko, P.: Semantic matching: algorithms and implementation. Journal on Data Semantics 9, 1–38 (2007)
11. Lee, M.L., Yang, L.H., Hsu, W., Yang, X.: Xclust: Clustering XML schemas for effective integration. In: CIKM 2002, pp. 63–74 (2002)
12. Madhavan, J., Bernstein, P.A., Rahm, E.: Generic schema matching with cupid. In: VLDB 2001, pp. 49–58 (2001)
13. Nayak, R., Tran, T.: A progressive clustering algorithm to group the XML data by structural and semantic similarity. IJPRAI 21(4), 723–743 (2007)
14. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB Journal 10(4), 334–350 (2001)
15. Saleem, K., Bellahsene, Z., Hunt, E.: PORSCHE: Performance oriented schema mediation. Information Systems 33(7-8), 637–657 (2008)
16. Tekli, J., Chbeir, R., Yetongnon, K.: An overview on XML similarity: background, current trends and future directions. Computer Science Review (2009)