Vol.4,No.1, July, 2015
178 - 185,ISSN:0975-5446] [VNSGU JOURNAL OF SCIENCE AND TECHNOLOGY]

Joint Word Segmentation based Heterogeneous XML Schema Matching Approach

JOSHI Dhaval A.
M.Sc.(I.T.) Programme,
Veer Narmad South Gujarat University
Surat, Gujarat, India.
joshi.dhaval@hotmail.com

PATEL S. V.
Dept. of Computer Science,
Veer Narmad South Gujarat University
Surat, Gujarat, India.
patelsv@gamil.com

Abstract

Data storage is always at center for any data centric application. Such data storage requires data intensive operations like migration, reshaping, restructuring, etc. to perform such operations; XML is widely used format because of its unique characteristics. XML schemas are heterogeneous as they are designed with different approach and purpose. Due to this heterogeneity, schema matching is a challenging task. Researchers have provided different techniques for schema matching, however no accurate technique exists to match schema elements which contain joint words. We have designed and implemented a new model for XML schema matching using joint word segmentation (segregation) technique which provides batter accuracy.

Keywords: XML Schema matching, joint word segmentation, syntactic, semantic, similarity measure.

1. Introduction:

Data storage is always at center and critical for any real world application and XML is one of the storage structures for such applications for data storage, data migration and web service message passing mechanism. Many applications may need to map data from heterogeneous XML storage for transferring data from legacy system to new system, OLAP modeling in data warehousing and data synchronization between heterogeneous applications. XML schema matching is the first step for data mapping.

XML schemas become heterogeneous as they are designed with different approach and purpose. So, it is difficult to match them and also becomes a very tedious task when schemas are so large. However experts can match them well, we always strive for more automation to make schema matching simple yet accurate. Many researchers have made good solutions for XML schema matching automation, but still need more accurate solutions with less human efforts. So, we

have designed and implemented a new model for XML schema matching using joint word segmentation (segregation) technique which provides batter accuracy.
The remainder of the paper is organized as follows. Section-2 provides background details of XML schema matching where it specifies types of matching techniques. Section-3 specifies related work which contains various techniques provided by different researchers for XML schema matching. Our model for XML schema matching is specified in Section-4. Experiments and results are described in Section-5. Section-6 concludes about the model described with experiments in this paper and future work for enhancement.

2. XML schema matching

XML schema matching can be done by various methods which can be categorized as follows:

2.1 Name Similarity Measure

Name similarity measure is based on two different types of measures, like syntactic measure and semantic measure. Syntactic measure gives result as two names are similar or not in terms of character sequence. If group of characters of one name is part of other name, it defines them as matching names. (e.g. Name and PersonName). But this measure is not capable to match semantic names. To match semantic names, Semantic matcher is required.
Semantic measure is based on Natural Language Processing techniques like synonym match, semantic distance match, etc. where words are compared with their semantic meaning. (e.g. trainee and apprentice).

2.2 Datatype Similarity Measure

XML element can contain any type of data though it is having same name in two different schemas or same data with different names in two schemas. Hence, it requires datatype compatibility to measure similarity. XML schema standard has 44 built-in types which also plays important role for XML schema matching.

2.3 Constraint Similarity Measure

When one schema element is based on the other schema element to define data value, it is a case of constraint similarity. Cardinality constraint is defined by minimum occurrence and maximum occurrence of an element that may appear in XML document. Using such parameters constraint similarity is measured.

2.4 Annotation Similarity Measure

In many XML documents, annotation is used to describe content of the element which is a full text description of the element. It can also be used to match elements using token-based similarity measures. In this measure, two different XML schema elements are compared with description content matching based on ranking with threshold value to generate matching result. To perform description content matching syntactic and semantic comparison plays major role.

2.5 Element Similarity Measure

XML element similarity is based on its characteristics like name, datatype, range, constraint, etc. To compare XML elements, one or more similarity measures as defined above can be used. Generally name similarity measure and datatype similarity measure are widely used by many researchers.

2.6 Structural Similarity Measure

XML document schemas are also compared with their structures. First element similarity is needed to get a matching element in two different XML schemas. Once two nodes are compared by element similarity, their ancestor, siblings and decedents are compared. Based on the comparison values, XML schema structures are finalized with matched and unmatched result.

3. Related Work

Thang and Nam [1], presented a solution for XML schema automated matching problem which produces semantic mappings between corresponding schema elements of given source and target schema which is based on combining linguistic similarity, data type compatibility and structural similarity of XML schema elements. Their solution contains two important tasks, like modeling XML schema and element similarity measure. In modeling XML schema, directed labeled schema graph with constraint sets is defined over both nodes and edges with the help of ideas presented in [2, 3, 4] which also classify schema graph nodes into atomic nodes and complex nodes. Each leaf node in the schema graph has atomic value (string, integer, date, etc.), list value or union value and each internal node in the schema graph has a complex content, which refers to some other nodes through directed labeled edges. They computed element similarity by combining linguistic similarity measure, datatype compatibility measure and structural similarity measure. For linguistic similarity, they combined two basic solutions among them one is presented in [2] and second is presented in [5] as Hirst and St-Onge algorithms with the concept of an allowable path to exploit WordNet. For datatype compatibility measure, they used a datatype compatibility table that gives a similarity coefficient between two given XML schema built-in datatypes, as defined in [2]. For structural similarity measure, node context matching is performed for two nodes using path similarity measure and context similarity measure. Their algorithm requires long computing time for schema matching as it passes through various similarity measures as well as their XML modeling is too complex for execution.

Dong and Linpeng [6], presented an ontology-based semi-automated technique to compare heterogeneous off-the-web XML schemas for integrating them. They constructed a layered approach with an intermediate model to reduce the complexity of ontology derivation for semantic similarity. They combined heuristic rules based technique and domain experts inputs to define interpretation of mapping information. Their layered model converts the XML schema document model to RDF schema conceptual model and from that using semantic mapping RDF schema mapped to OWL global ontology. The final global ontological schema is constructed with heuristic approach for improving schema integration, used for data instance integration process to centralize all XML documents data for making centralized data repository. From various experiments, they found that the data instance population time increases approximately linearly with the size of input XML instance document and their implementation is capable to handle such files in an acceptable time. Their solution requires more enhancements to accommodate updates of the XML instance document and it is semi-automated which requires domain expert's input.

Khalid, Zohra and Hunt [7], introduced a technique for large scale schema matching using tree mining. They investigated scalability with respect to time performance in the context of approximate mapping where tokenization, abbreviations and synonyms were used for the linguistic matching of node labels. The matching strategy was hybrid and optimized for schemas in tree format. In their technique, they labeled each node and assign values to it, which is complex formation of tree and also need more computation for schema matching.
Above all techniques are not matching multi-named elements in schema matching. So here, present a new model which will be capable enough to match multi-named elements in large XML schema matching as detailed in section IV.

4. XML Schema Matching Model using Joint Word Segmentation

Our XML schema matching model is depicted in Figure.1 and all phases are explained along with examples.

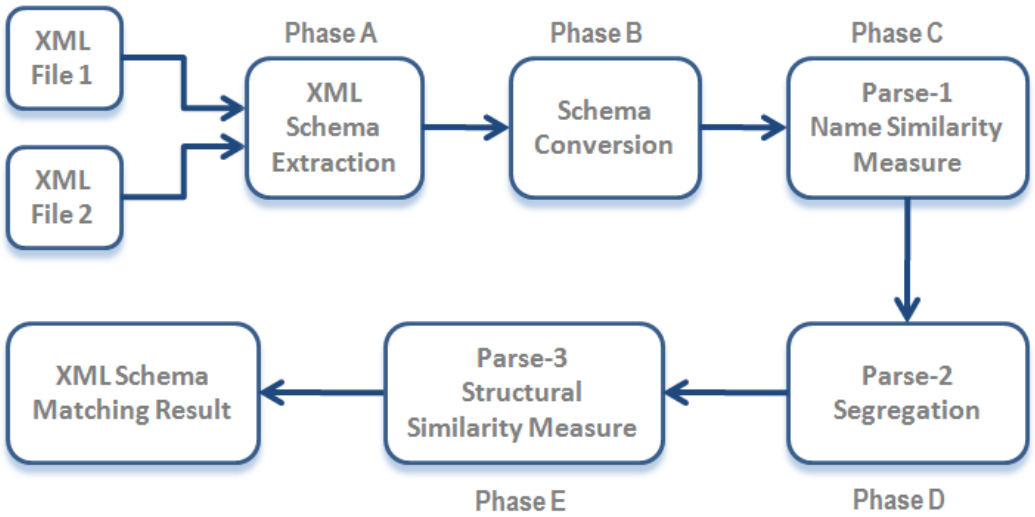


Figure 1: XML Schema Matching Model using Joint Word Segmentation

A) XML Schema Extraction

To compare schema of two XML documents we need to extract schema contents from it and that contents are passed on to the next phase for systematic text formation as specified in phase (B).

B) Convert Schema to text representation with specified separator.

The extracted schemas in phase (A) of both XML documents need to be converted to the specified text with predefined separator such that all schema elements can be identify with complete path of the document. This path identification structure can be used for comparison of two different XML elements by following phase (C) to (F) by parsing the list with various comparison techniques.

e.g. Book* Title

C) Parse – 1: Use name similarity measure for comparison of names from top to bottom and put match contents in different list.

In this phase, text representation generated by phase (B) is used for comparison of elements using name similarity measure. For every pair of elements from two XML schemas first syntactic similarity measure can be used and then on the remaining unmatched pair of elements semantic similarity measure can be used. For semantic similarity we use WordNet and Watson [8,9] open source API. Each compared elements pair identified in this parse is removed from lists which were generated in phase (b). The remaining lists are passed to the next phase (D) for further comparison.

e.g. name -> username (Syntactic similarity)

e.g. student - > pupil (Symantec similarity)

D) Parse – 2: (Segregation) Decompose words wherever possible, use name similarity measure to compare from top to bottom and add matched results in different list.

The phase (C) can't identify joint words which are not compared by syntactic similarity and for that decomposition of joint words needs to be performed for comparison. For decomposition of element name, we use m-gram algorithm to get combination of words less than joint word. Each identified word is checked using dictionary and meaningful word is only used for matching using name similarity measures as defined in phase (C).

e.g. DepartmentId -> DeptID and EmpCodeID -> EmployeeCodeIdentification

E) Parse – 3: Based on the Parse 1 to 2 results, use structural similarity to compare remaining list.

In this phase, the compared list and non-compared lists can be used for structural similarity. From the compared list, select a pair and search for the other related elements from non-compared lists and element similarity measure can be used to match those elements. Related elements from the non-compared lists can be considered as parent, child, sibling, inner attribute, etc.

So it can be seen that with the help of this model we will be able to match XML schema elements more accurately as follows:

name -> username
student -> pupil
DepartmentId -> DeptID
EmpCodeID -> EmployeeCodeIdentification

Hence, we can say that problem of multi-named elements matching can be effectively solved and schema matching performance can be improved using above methodology.

5. Experiments and Results

We used three sets of heterogeneous XML schemas to test accuracy of our model. One of the sample sets is shown in Figure–2 and Figure–3. Figure–2 contains Institute Management and Figure–3 contains School Management schemas in tree structure. Moreover XML structure representations of both schemas are shown in Figure-4. Both schemas are able to store similar information but their structures are quite different. Schema-A contains identifiers as attributes and schema-B contains identifiers as elements.

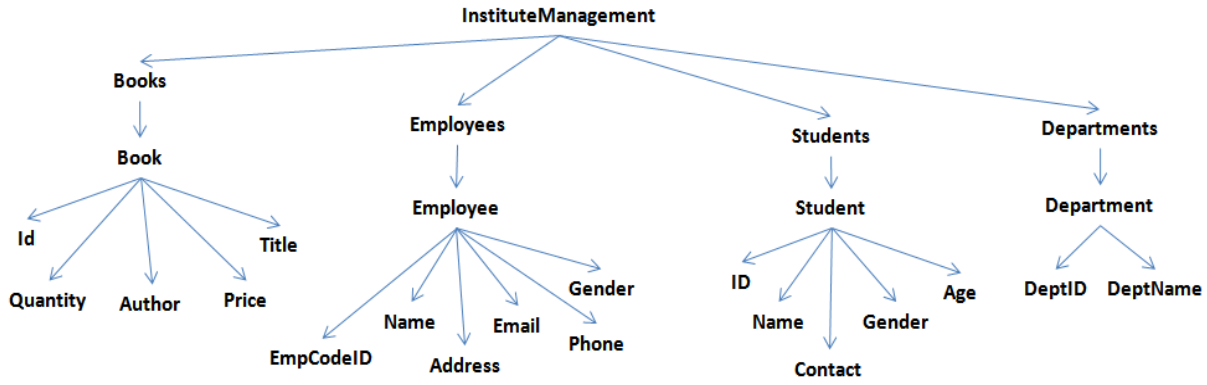


Figure 2: Institute Management Schema

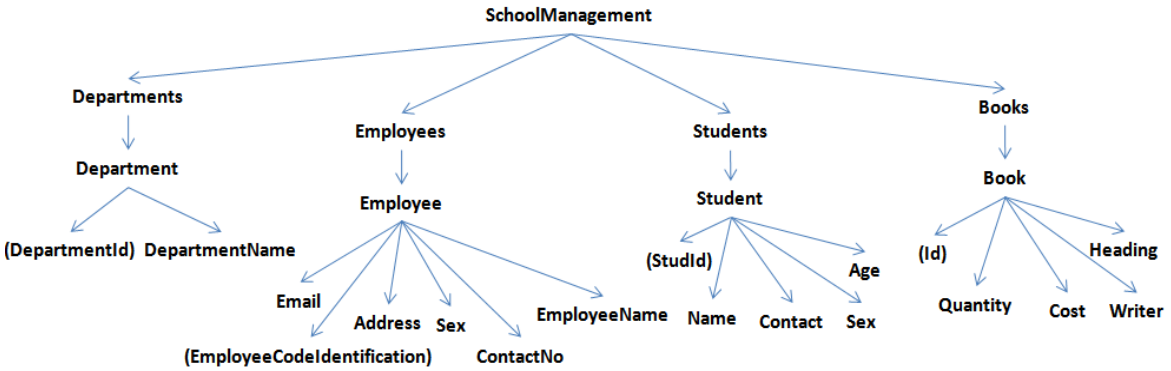


Figure 3: School Management Schema

```
<?xml version=”1.0” encoding=”UTF-8”?>
<SchoolManagement>
   <Employees>
      <Employee
EmployeeCodeIdentification=””>
```

```
<?xml version="1.0" encoding="UTF-
8"?>
<InstituteManagement>
   <Books>
      <Book>
```

```
        <EmployeeName></EmployeeName>
        <Sex></Sex>
        <Address></Address>
        <ContactNo></ContactNo>
        <Email></Email>
      </Employee>
    </Employees>
    <Books>
        <Book Id=””>
            <Heading></Heading>
            <Writer></Writer>
            <Cost></Cost>
            <Quantity></Quantity>
        </Book>
    </Books>
    <Departments>
      <Department DepartmentId="">

<DepartmentName></DepartmentName>
      </Department>
    </Departments>
    <Students>
      <Student StudId="">
        <Name></Name>
        <Contact></Contact>
        <Sex></Sex>
        <Age></Age>
      </Student>
    </Students>
</SchoolManagement>
```

Schema (A)

```
      <ID></ID>
      <Title></Title>
      <Author></Author>
      <Price></Price>
      <Quantity></Quantity>
    </Book>
  </Books>
  <Students>
    <Student>
      <ID></ID>
      <Name></Name>
      <Gender></Gender>
      <Contact></Contact>
      <Age></Age>
    </Student>
  </Students>
  <Employees>
    <Employee>
      <EmpCodeID></EmpCodeID>
      <Name></Name>
      <Gender></Gender>
      <Address></Address>
      <Phone></Phone>
      <Email></Email>
    </Employee>
  </Employees>
  <Departments>
    <Department>
      <DeptID></DeptID>
      <DeptName></DeptName>
    </Department>
  </Departments>
</InstituteManagement>
```

Schema (B)

Figure 4: Sample set of XML Schemas for Comparison

- After passing both the XML schemas to our model for matching and passing through different phases we receive matching results. Elements of schema-A like writer, cost, sex, heading, etc. are matched with elements of schema-B like author, price, gender, title, etc. respectively by Phase (C) – Name Similarity Measure.
- As Phase (C) can't match joint words elements like DepartmentID, DepartmentName, EmployeeCodeIdentification, etc. of schema-A with DeptID, DeptName, EmpCodeID, etc. respectively of schema-B, they are passed to the Phase (D) – Segregation for comparison. Phase (D) produces matching results as shown in Table-1 by using segregation technique.

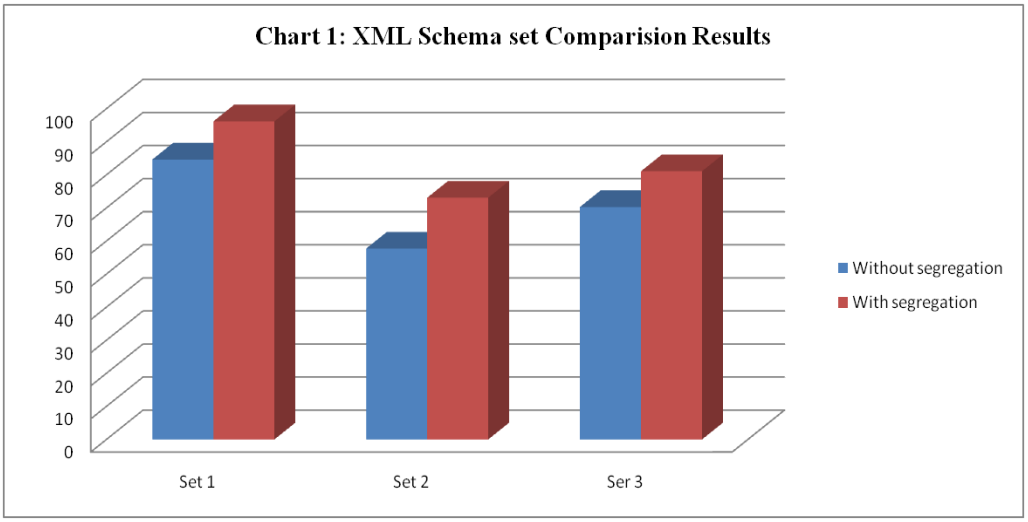
Table 1: Matching result based on Parse-2 (Phase D) – Segregation of both schemas

SchoolManagement Schema	**InstituteManagement Schema**
Departments/Department/DepartmentId	Departments/Department/DeptID
Departments/Department/DepartmentName	Departments/Department/DeptName
Employees/Employee/EmployeeCodeIdentification	Employees/Employee/EmpCodeID

Hence, in the view of the above (Table-1), combined results generated due to Phase-C and Phase-D, are quite accurate. Similarly we tested our model with many schema sets. Results of few of them are shown in Table-2. Also matching accuracy results are shown graphically in Chart-1 which also indicates that schema matching with segregation improves schema matching results for joint word elements of schemas.

Table 2: Matching result accuracy by model for three different sets of schemas

XML Schema Sets	Matching accuracy in %	
	Without Segregation	With Segregation
Set 1	84.61538462	96.15384615
Set 2	57.69230769	73.07692308
Ser 3	70.27027027	81.08108108



6. Conclusion and Future Work

XML schema matching automation is always a challenging task. We presented a model for heterogeneous XML schema matching using joint word segmentation technique which is slow compare to individual measure based techniques. However, as our model focuses on specific aspects of joint word element schema matching, it generates more accurate matching results.
As an extension to the above work, we need to enhance the model for multi-level structural uncertainty of heterogeneous XML schemas.

References

[1] Huynh Quyet Thang, Vo Sy Nam, XML Schema Automatic Matching Solution, International Journal of Electrical, Computer and System Engineering 4:1, 2010, World Academy of Science, Engineering and Technology, Vol:4 2010-03-24, International Science Index Vol:4, No:3, 2010, waset.org/publications/3391

[2] J. Madhavan, P. A. Bernstein, and E. Rahm. Generic schema matching with Cupid. MSR Tech. Report MSR-TR-2001-58, 2001, Available at: http://www.research.microsoft.com/pubs.

[3] S. Melnik, H. Garcia-Molina, E. Rahm. Similarity Flooding: A versatile Graph Matching Algorithm and its Application to Schema Matching. In Proceedings of the 18th International Conference on Data Engineering, 2002. Available at: http://dbpubs.stanford.edu/pub/2001-25. (Extended Technical Report, 2001.)

[4] A. Boukottaya, C. Vanoirbeek. Schema Matching for Transforming Structured Documents. In DocEng'05, 2-4, 2005.

[5] A. Budanitsky and G. Hirst. Semantic distance in WordNet. An experimental, application oriented evaluation of five measures, 2003.
[6] Li Dong, Huang Linpeng, A Framework for Ontology-based Data Integration, International Conference on Internet Computing in Science and Engineering (ICICSE'08), Harbin, Page(s): 207-214, ISBN: 978-0-7695-3112-0, DOI: 10.1109/ICICSE.2008.96, 2008 IEEE Explore.
[7] Khalid Saleem, Zohra Bellahsene, Ela Hunt, Performance Oriented Schema Matching, DEXA'07 Proceedings of the 18th international conference on Database and Expert Systems Applications, Pages 844-853, ISBN:3-540-74467-3 978-3-540-74467-2, 2007
[8] Jérôme Euzenat, Watson, more than a Semantic Web search engine, Semantic Web 0 (0) 1, IOS Press, http://www.semantic-web-journal.net, 2011
[9] Mathieu d'Aquin, Claudio Baldassarre, Laurian Gridinoc, Marta Sabou, Sofia Angeletou, Enrico Motta, Watson: Supporting Next Generation Semantic Web Applications, The Open University's repository of research publications and other research outputs, 2007