A Comparison of Algorithms to Construct Ontologies from Relational Databases

Sameen Fatima
Faculty of Computer Science
Institute of Business Administration
Karachi, Pakistan
e-mail: shigon_sl@hotmail.com

Quratulain Rajput
Faculty of Computer Science
Institute of Business Administration
Karachi, Pakistan
e-mail: qrajput@iba.edu.pk,
quratulain.rajput@gmail.com

***Abstract*—Relational databases (RDB) play a vital role in managing the organization's data but they are dependent on autonomous hardware and software and thus create a problem of data integration. On the other hand Ontologies are considered as one of the most popular solutions in knowledge representation as a universal language to share and integrate knowledge. To overcome the integration problem in database and to realize the vision of semantic web, data has to publish over a web as ontologies. The purpose of this research is to explore algorithms to construct ontologies automatically from relational databases. A comparison is made on the basis of degree of automation and accuracy to transform relational database into ontology. Finally, assessing the strength and weakness of each algorithm and explore the future research directions.**

Keywords- Ontology; Relational database; Mapping rules

I. INTRODUCTION

It is said that Information is power. If the information is separated and isolated from other data it cannot bring any value to the organization. Before the emergence of database systems it was difficult to manage this information. As the organizations engaged their resources in managing a lot of duplicated data, handling data dependencies, dealing with incompatible file formats and representation of data from user's view, they cannot utilize this information to its full potential. Database systems were introduced to manage these autonomous files as a single centralized collection of data. These systems reduce data duplication, avoid data inconsistency, allow sharing of data, increased security and maintain data integrity and make it available on demand [1]. This approach remained successful and sufficient to meet user requirements for several years. However, today's user data processing requirements and capabilities have changed and new applications often involve accessing and maintaining data from several pre-existing databases, which are typically located on autonomous software and hardware platforms distributed over the many sites of a large computer network which leads to heterogeneity and legacy problems, initiating a need for timely and efficient solution by sharing existing knowledge [2].

Besides ongoing advances in database technologies there are still the challenges of uniform and scalable access to multiple information sources including databases and other repositories [3]. Now, (World Wide Web) WWW is playing a more vital role in information sharing for the purpose of education, business, research etc. therefore more and more people are publishing the data over web to share it among large audiences. However, Data is being published in different formats such as PDF, Doc, HTML, etc. Among these different formats of data, most of the information is coming from databases. One of the study reported that "it was determined that Internet accessible databases contained up to 500 times more data compared to the static web and roughly 70% of websites are backed by relational databases"[4]. With the continuous increase in the volume of published data, it is desirable to provide some automatic mechanism to search and integrate information over the Web which is not possible on the existing web. In recent years, with the advent of semantic web technologies (RDF [5], RDFS [5], and OWL [5]) that have been standardized under W3C group, has proven to be a powerful support for the techniques used for managing data and for the problems of data heterogeneity and semantic interoperability [5]. Ontologies (RDFS or OWL) have been suggested as a way to solve the problem of information heterogeneity by providing formal, shared and explicit definitions of data called semantics. The addition of such semantics also improves the query processing by providing more meaningful answer. Additionally, ontologies also have reasoning ability to infer new knowledge and to identify inconsistencies. An ontology-based access to relational data reduces the barriers for data exchange and integration. The expressive and formal semantics increases the value of the existing data and enables new applications on that data [3].

Recently different projects have been developed over Web using semantic web technologies such as DBpedia, Semantic wikis. Moreover, due to the popularity of ontologies, now commercial relational databases (such as Oracle) also provide support of ontologies. However, the construction of ontology is still manual [6]. Thus, it is highly desirable to transform databases into ontologies mainly because of two reasons, first to publish relational data as RDF/OWL on the web and secondly to combine a relational data with existing RDF/OWL for data integration.

This paper compares the existing work by comparing three different algorithms to automatically/semi-automatically construct ontologies from relational databases that can provide a conceptual view over the data. Therefore we can take advantage of both technologies. To construct the

 ISBN: 978-1-61208-202-8

ontology model these algorithms established rules between ontological constructs (concept, relation, individual, etc.) and relational databases (tables, attributes, attribute values, etc.). Recently, several approaches have been proposed in literature to transform databases into ontology [7][8][9][10].

The rest of the paper is outlined as follows. The next section presents a brief overview of three recently proposed algorithms to construct ontology from Relational databases with example. Section III compares these algorithms to indicate the challenges involved in this research. Finally, Section IV provides conclusion and future research directions.

II. ALGORITHM FOR ONTOLOGY CONSTRUCTION FROM RELATIONAL DATABASES

The construction of ontologies from relational databases and the development of such type of tools has been a major field of interest of the researchers with the evolution of semantic web [11]. In this paper, we have selected three most recently proposed algorithms and describe each algorithm separately with example to investigate the challenges in this field. In general, following are the main components of relational database and ontology that are being considered in the selected algorithms.

Components of Relational database:

- Relations (Tables)
  - Entity tables
  - Relationship tables
- Attributes of relation
  - Key attributes (Foreign key, Primary key)
  - Non-key attributes
- Restriction
  - Limit on the attribute value
  - UNIQUE value attribute (Primary key)
  - NOTNULL (attribute value cannot be null)

  …

Components of ontology:

- Concepts/Classes
- Relationships
  - Taxonomy relation (Class hierarchy)
  - Non-Taxonomy relation (Object and Data type properties)
- Restrictions (Axioms)
  - Cardinality restrictions
  - Functional property

  …

The above mentioned components are analyzed to identify the associations between ontological and relational component. These associations would result in the development of rules to construct ontology automatically from relational databases. Furthermore, to understand the working of rules of each algorithm an example of relational database has been selected as shown in Figure 1.

Student

StudId	StudName	TownId
S001	Saad	T311
S004	Kashif	T119
S102	Faisal	T108

Department

DeptId	DeptName
D008	Computer Science
D119	Electronics
D203	Mathematics

Town

TownId	TownName	TownPcode
T311	Gulshan	73500
T108	Sadder	73400
T119	Defence	77400

PhDStudent

StudId	Reseachfield
S001	Database
S004	Semantic Web
S009	Networking

Employee

EmpId	EmpName	EmpSex
E003	Ahmed	male
E112	Sana	female
E203	Sadaf	female

Studies

StudId	DeptId
S001	D008
S004	D008
S102	D203

Affiliates

EmpId	DeptId
E003	D008
E112	D119
E203	D203

Figure 1: Example of relational database

A. Algorithm 1

This algorithm [7] was presented by Peng et al. emphasizes the problem of efforts and cost involved in the manual construction of ontologies. They suggest reducing the cost through ontology learning of structured data. The relational databases (RDB) come under structured data which is a domain specific model. To construct ontology they established correspondence rules between components of ontology and RDB where the tuple in the table shows the instance of Relational schema.

This algorithm divides the relations (tables in RDB) into two types of relations one is Correlative Relation and another one is Basic Relation. Correlative relations are those which do not have any non-key attributes. In contrast, the relations which are not correlative are considered as Basic relations. Following are the rules developed to construct ontology.

Rules for Ontology Concept/Class:

- If a relation is a basic relation then it will be converted into class of ontology. For example, the database shown in Figure 1, tables of Student, PhdStudent, Town, Department and Employee are created as classes in ontology.

Rules for Ontology Relationships:

- If a relation is a correlative relation then it will be converted into two Object properties in ontology that show relation between entities. Example, consider the

Studies and affiliate tables of the database as shown in Figure 1 two Object properties will be created for each relation where one is inverse of the other.

- A primary key of correlative relation will be converted into Object property with their referenced tables. For the database example, stuId in table Studies.
- If a relation attribute is not a foreign key attribute then it will become Data type property in ontology. For example, consider the database example of Figure 1, the attributes of the relations such as StudName in relation Student, TownName and TownPcode in relation Town ResearchField in relation PhdStudent, and so on are created as Data type property in ontology.
- All the foreign keys in a basic relation will also create a relation of Object property. For a given example, attribute TownId is a foreign key in Student relation, thus create an Object property corresponding to TownId between classes Student (as domain) and Town (as range).

Rules for Ontology Restrictions:

- If a relation is a basic relation and has only one Primary Key then it will be converted to the data type functional property with minCardinality or maxCardinality equals to 1. For the example in Figure 1 consider primary keys of each basic relation.
- If a relation is a basic relation and has more than one Primary Key then each primary key will have a restriction with minCardinality or maxCardinality equals to 1.
- If the property is set as NOTNULL then it will take the restriction minCardinality as 1.
- If the attribute of relation is set as UNIQUE then restriction will be created as functional property.

Thus, the algorithm applied rules discussed above by getting MetaData of each table and construct ontology for relational database.

B. Algorithm 2

This algorithm [8] was presented by Zdenka Telnarova, like previous paper this paper also discusses the importance of automatic construction of ontologies from relational databases.

This algorithm transforms relational model in to conceptual model (ontology) by considering the reverse of transformation rules used in the transformation of conceptual model (ERD) to relational model (RDB). This algorithm proposed following set of rules that transform relational model into conceptual model (ontology).

Rules for creating Classes:

- If we have multiple relations in database and all of those relations have same Primary key then it is possible to integrate all of these relations under the single class/concept of ontology. This rule expresses that same primary key in different relation corresponds to same entity and for the purpose of normalization it was divided into more than one relation. Thus integration is a reverse of normalization. Consider an example shown in Figure 1 where primary key of student is also a primary key of phdStudent relation thus these two relations can be integrated into a single concept student in ontology.
- If we have relations in the database and no other relation could be integrated with it according to rule 1. Moreover, the attribute is primary key but not a foreign key in a given relation then this relation is created as a concept in ontology. Relations Town, Department, and Employee are example of such relations.

Rules for creating Relationships:

- The two relations, where attribute of one relation is equal to attribute of another relation and at the same time the common attribute is not the primary key in one relation (similar to foreign key), then Object property can be created. This rule reveals that to incorporate relationships between entities, foreign keys are being added in RDB thus it can be created as Object properties in ontology. For example TownId in student is created as Object property where domain concept is Student relation while range is Town concept.
- If we have two relations, then it is possible to create two Object properties if the following two conditions are fulfilled: a) a Relation has more than one Primary Key (as in relationship table) and b) Foreign Key of one relation belongs to Primary Key of another relation (similar to correlative relation according to algorithm 1). This rule is used to convert many to many relationships in RDB. For the database example two Object properties are created corresponding to studies relation where one property is an inverse of other property.
- All the other attributes of relations in database which cannot be converted into Object property according to the above rule become Data type property in ontology. For example StudName, ResearchField, TownName, TownPcode, EmpName, EmpTitle DeptName are created as Data type property.
- If we have two relations and have same primary key in both relation as well as it is also a foreign key in one relation. To express is-a relationships in RDB same identification key is being used in both super-type and sub-type, thus same identification key in different relation indicate hierarchical relationship which can be exploited to create hierarchy of concept in ontology.

 ISBN: 978-1-61208-202-8

For example, student with primary key stuId and phdStudent with primary key stuId which is also a foreign key. Therefore phdStudent become a subclass of student.

Rules for Restrictions:

- If the attribute in a relation is the primary key then it will have restriction with minCardinality and maxCardinality equals to 1.
- If the attribute in the relation is declared as NOTNULL then the minCardinality equals to 1.
- If the attribute in the relation is declared as UNIQUE then the maxCardinality is equals to 1.

Rules for creating Instances:

- If C is the corresponding class to database relations R_1 or integration of more than one relation then each tuple is considered as an instance of Class C.

C. Algorithm 3

This is a recently proposed algorithm [9] that has been proposed by Zhou et al., to construct ontology for relational database and used WordNet to further extend/reorganize the ontology. Similar to algorithms 1 and 2, this algorithm also described the rules to obtain ontology for a given database.

This algorithm also divides the table in to two types of table one represent entity and one represent relationship between entities. They also suggest the finding of inheritance between concepts.

Rules for creating Classes:

- If the table is entity table (similar to basic relation in algorithm 2) then create corresponding concept/class in ontology. As the example shown in Figure 1, studies and affiliates are relationship tables while others are entity tables for which concepts are created.

Rules for creating relationships:

- If table T1 is a relation table, T2 and T3 are entity tables that correspond to C2 and C3 concepts in ontology then two Object properties will be created; in which one is the inverse property of other. For example, Studies is transformed in to two Object property between concepts Student and Department.
- If a column is not a foreign key, it will be transformed to the Data type property of related table.
- If table T1 has T2's foreign key, then foreign key will be transformed into an Object property, its domain is T1's corresponding concept, range is T2 corresponding concept. If a column is not a foreign key in a relation table, it will become the common property of related concepts.
- If there is a column within an entity table that could have several values, no matter how many records, some sub concept could be created by the column's data value. For example, in Employee table, EmpSex has only two possibilities male/female. Thus two sub-concepts can be created such as maleEmployee and femaleEmployee.

Rules for creating Restrictions:

- An entity table with attribute declared as NOT NULL then the corresponding Data type property is restricted to minCardinality equal to 1. However, a foreign key and a primary key corresponded data type properties has minCardinality and maxCardinality are both set as 1.

Rules for creating Instances:

- Each record of entity table will be transformed into related concept's instance.

Once the ontology has been created by applying above rules, next, this algorithm will use WordNet to extend the ontology. The extension has been done in two ways. First, by adding concepts synonyms, for example, the synonym of department is section, thus, anyone can be used to refer the concept. Second is hierarchy extension, in case of more than one database some of the concept may be found parallel thus by checking the hyponymy to modify the created ontology.

Figure 2 shows the ontology created in general by applying above algorithms on database example shown in Figure 1. In Figure 2, oval shapes represent concepts in ontology, solid line rectangles represent Object properties; dotted rectangles represent Data type properties while head of dotted arrows indicate range and tail indicate domain of properties. However the solid arrow represents is-a relationship between concepts.

III. COMPARISON OF ALGORITHMS

This section analyzes the research challenges in ontology construction from relational databases by comparing algorithms as explained in the previous section. These algorithms are compared to identify issues involved in the construction of new ontology in general and the issues specific to each algorithm. In general, the rules to construct ontology components are based on the design of relational models. These algorithms exploit the semantics of relational database components such as entity table, relational table, attributes, and constraints, etc. The algorithms can be divided in to three categories, first is the construction of ontology from database schema, second is the construction of domain specific ontology and third is the mapping of database with existing ontology [12]. In domain specific ontology, the

 ISBN: 978-1-61208-202-8

purpose is to select data that is relevant to the domain rather than to create mirror of database as ontology. The algorithms discussed in this paper come under the first category. Following subsections elaborate the comparison of the selected algorithms.

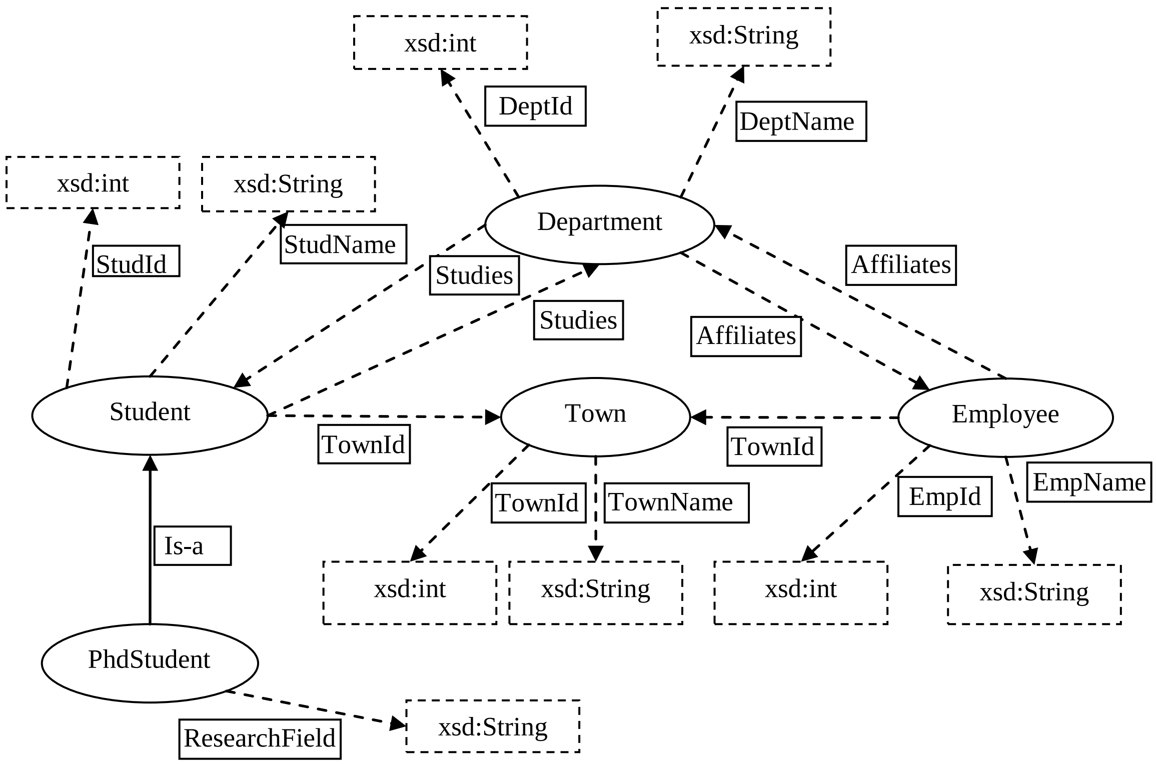

Figure 2: Ontology created from database

A. Motivation and purpose stated by authors

The motivations highlighted by authors to construct ontologies from databases are: (i) generate data for semantic web that is processable by machine, and (ii) integration of data from different sources (heterogeneous databases or existing RDF/OWL). To create data for semantic web, databases are rich resource of information and it has been found that large amount of information in dynamic web pages are also generated from databases. Therefore rather than to annotate each dynamic web pages, a better solution would be to create ontology corresponding to a database. Once the ontology has been created it can be used to generate semantic web content that can be processable by machine. Furthermore, it is highly desirable to integrate data from heterogeneous databases or existing RDF/OWL to process the data at large scale for different application development. The authors suggest the use of ontology as the ontologies provide shared and reusable knowledge representation as universal model that support in data integration. Therefore, algorithms are needed to construct ontology from database as discussed in this paper.

B. Algorithms automation level

The automation is an important factor to perform transformation from RDB to OWL over a large scale. Therefore authors preferred higher degree of automation in their algorithms using mapping rules. However, mapping rules are defined manually. Once the ontology has been created some post processing is required to formalize the domain vocabularies used in ontology. Furthermore, all three algorithms are limited to basic mapping rules. Addition to this some additional mapping rules need to be defined to extract more semantic information from relational database to be filled in ontology. To extend the use of ontology Peng et al., suggested that the construction of these local ontologies further extended by providing mapping between ontologies and this process could be completely automated.

C. Algorithm's Ontology language

This section refers to the selection of appropriate language to represent the ontology. In spite of the fact that there is a demand to transform RDB into OWL or vice versa, there is no standard language has been developed specifically for representing the mapping between RDB and ontology [12]. Therefore, all three algorithms discussed in this paper used ontology languages, such as RDFS/OWL. It has been noted that the use of RDFS is sufficient for components of ontology that are currently being filled by these algorithms. However, the use of OWL provides additional components (not in RDFS) that might require in future or need to fill manually depending on the application.

D. *Algorithms Implementation*

This section indicates that either these algorithms provide software/tool to practically perform RDB to ontology conversion so that potential user can take benefit from it. Algorithm 1 has been implemented and a prototype is created to perform experiment on Oracle 10g, SQL Server 2000, and MySQL Server 5.0 databases, however, experiment results have not been provided. Algorithm 2 and Algorithm 3 did not provide information about implementation of their algorithms, whereas explained methodology with examples.

E. *Issues and Challenges*

The algorithms used basic transformation method by creating mapping rules between ontology and relational database components such as table to class, column to predicate etc. These basic mapping rules are generic to apply transformation over a large scale without human involvement. However, database schema is not sufficient information to generate corresponding ontology. Therefore semantically rich ontology can be generated by gathering information from table data, queries, and stored procedures.

The standard database design is based on conceptual model which is then transform into relational model. On the other hand ontology is a conceptual model therefore algorithms develop mapping rule by considering the reverse transformation i-e relational model to conceptual model. However, these reverse transformation most of the time are not possible because databases that have lost original intention in the transformation (conceptual model to relational model) and are very difficult to reverse back. Moreover, most of the transformations in databases development are not documented at all.

F. *Recommendations*

All three algorithms discussed in this paper highlighted the importance of automatic ontology construction from relational databases and focuses on building domain specific ontologies. This section describes the recommendations and future direction provided by the authors of selected algorithms. One possible recommendation would be to create ontology from conceptual model of database (ERD or UML) and using queries extract data from database to populate into ontology. Secondly, the newly constructed ontology can be further extended by adding more semantics into it and several such ontologies can be integrated to share or exchange knowledge [7]. Moreover, these ontologies would help in building of knowledge warehouse to further extend their utilization [8].

IV. Conclusion and Future Direction

Database interoperability and semantic reasoning is the ultimate target that the researchers are trying to solve. This considers allowing different databases to be semantically integrated. The paper described importance of ontology construction from database and discussed three recently proposed algorithms. More specifically, these algorithms defined some rules to build a generic approach without human involvement. In spite of all the efforts have been done so far only basic mapping rules are investigated.

Future research directions would focus in extension of the basic mapping rules by adding more mapping rules to create semantically rich ontology.

References

[1] Te-Wei Wang and Kenneth E. Murphy, “Semantic Heterogeneity in Multidatabase Systems: A Review and a Proposed Meta-Data Structure”, Journal of Database Management, Vol. 15, No. 4, pp. 71-87,2004

[2] Natalya F. Noy, “Semantic Integration: A Survey Of Ontology-Based Approaches”, SIGMOD Record, Vol. 33, No. 4, pp. 65-70, 2004

[3] Matthias Hert, University of Zurich, Binzmuehlestrasse 14, CH-8050 Zurich, Switzerland hert@ifi.uzh.ch, “Relational Databases as Semantic Web Endpoints”

[4] B. He, M. Patel, Z. Zhang, and K. C.-C. Chang, “Accessing the deep web,” *Commun. ACM*, vol. 50, no. 5, pp. 94–101, May 2007.

[5] T. Berners-Lee, J. Hendler, and O. Lassila. “The semantic web, a new form of web content that is meaningful to computers will unleash a revolution of new possibilities”. Scientific American, 2001.

[6] Oracle documentation, docs.oracle.com, last access March 25, 2012.

[7] Peng Liu1, Xiaoying Wang, Aihua Bao, Xiaoxuan Wang,”Ontology Automatic Constructing Based on Relational Database”, 2010 Ninth International Conference on Grid and Cloud Computing

[8] Zdenka Telnarova, “Relational Database as a source of ontology creation”, Proceedings of the International Multiconference on Computer Science and Information Technology, pp. 135-139, 2010 IEEE

[9] Xu Zhou, Guoji Xu, Lei Liu, “An Approach for Ontology Construction Based on Relational Database”, International Journal of Research and Reviews in Artificial Intelligence, Vol. 1, No. 1, March 2011,Copyright © Science Academy Publisher, United Kingdom

[10] W3C RDB2RDF Incubator Group,“A Survey of Current Approaches for Mapping of Relational Databases to RDF” ,January 2008-09

[11] Man Li, Xiao-Yong Du, Shan Wang, “Learning Ontology from Relational Database”, Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005 .

[12] Dimitrios-Emmanuel Spanos, Periklis Stavrou, and Nikolas Mitrou, “Bringing relational databases into the semantic web: A survey,” *Semantic Web*, 2010.