Automatic Ontology Learning from Heterogeneous Relational Databases: Application in Alimentation Risks Field

Aicha Aggoune

Automatic Ontology Learning From Heterogeneous Relational Databases: Application in Alimentation Risks Field

Aicha AGGOUNE

Department of Computer Science, LabSTIC Laboratory, University of 8th May 45
Guelma, Box 401, Algeria
aggoune_ai@yahoo.fr

Abstract. In this paper, we propose a semantic approach for automatic ontology learning from heterogeneous relational databases in order to facilitate their integration. The semantic enrichment of heterogeneous databases, which cover the same domain, is essential to integrate them. Our approach is based on Wordnet and Wup's measure for measuring the semantic similarity between elements of these databases. It is described by a detailed process that can allow not only the generation of ontology but also its evolution as the evolution of its databases. We applied our approach in the alimentation risks field that is characterized by a large number of scientific databases. The developed prototype has been compared with similar tools of generation ontology from databases. The result confirms the quality of our prototype that returns the generic ontology from many relational databases.

Keywords: Ontology Learning, Ontology Evolution, Relational Databases, Wup's Similarity Measure, Wordnet.

1 Introduction

Ontologies do not replace relational databases (RDBs) but offer an alternative to RDBs to provide the meaning of the data and so they can facilitate the data integration. Different approaches for ontology building such as ontology building from scratch, ontology learning from text, ontology learning by reusing of existing Ontologies, etc [8]. The ontology learning has since emerged as an important domain of ontology engineering that consists to generate an ontology from various sources such as text, database, dictionary, etc [16]. The ontology learning from heterogeneous relational databases is the scope of this paper. We focus on two kinds of heterogeneity: the structural heterogeneity and semantic heterogeneity. The structural heterogeneity is related to different representations of data via tables, the attributes, and the relations, which represent themselves by tables, but the semantic heterogeneity corresponds to the different definitions to describe the same data [1].

In our previous work [1], we have proposed a novel semantic mediation system for solving semantic heterogeneity at both the query and database levels. The proposed system is based on hybrid ontology approach, which consists in attributing to each database its own ontology, called local ontologies describe its knowledge.

In the context of semantic integration of heterogeneous databases, we attempt to present in this paper a novel approach to automatic learning of domain ontology from relational databases, which cover the same domain rather than built from each database its own ontology. Using one domain ontology of many relational databases allows improving the performance by reducing the execution time and space memory occupied by the ontology schema. To validate our approach, we have chosen the alimentation risks field that is characterized by a large number of scientific databases.

This paper is divided into five sections. Except for the introduction presented in the first section, the second one devoted to the works in the literature, which are related to automatic ontology generation from relational databases. Section three gives in detail our semantic approach. Section four describes the experiments and results. In section five, we give conclusion with future work.

2 Related Work

Ontologies represent a primordial semantic web technology that serves as a standard vocabulary for the sharing and reusability of knowledge [3]. They are used to improve data representation by associating well-defined meaning with data. Thus, the ontology is used in different topics related to semantic databases such as semantic enrichment for database [11], disambiguation data [20], semantic data integration [1], etc. Using ontology as a key element for dealing semantic conflicts in databases according to different ways; either for creating database from ontologies [10], ontology learning from database [14], or managing data and making decisions [7]. In this paper, we are interested in the second way in order to propose a new approach for automatic ontology learning from relational databases. In this area, different approaches have been proposed for only generating an ontology from a single database [4], [13]. Upadhyaya et al. [19] proposed an algorithm to transfer extended Entity/Relationship diagram (E/R Diagram) to OWL ontology. The algorithm is semi-automated because that requires a domain expert to aid more meaningful information and obtain a richer ontology. Fahad [9] proposes a framework for transforming the structured analysis and design artifact, E/R diagram, into the OWL ontology. This framework is limited to handle other cases of relationships between entities that are not binary, which require reification.

Recently, Dadjoo and Kheirkhah [5] proposed an approach for automatic ontology construction based on the relational database. This approach is based on graph theory, which leads to product the graph from database, and also with transforming of the graph obtained, final the ontology has been generated. This approach has succeeded

to show richer semantics in the target ontology but it is not suitable for ontology construction from heterogeneous databases.

In addition, there are several tools allowing mapping relational databases (RDBs) to ontologies. Some of the most notable tools are DataMaster [19], KAON2 [11] and RDBToOnto [12]. These approaches define the mapping between components of both database and ontology, which an ontology class corresponds to a RDB table, an ontology datatype property corresponds to a table field, an ontology object property corresponds to an RDB attribute, and an ontology class instance corresponds to a RDB record [4].

The aforementioned tools are limited to the homogeneous database and they cannot ensure the ontology evolution as the heterogeneous databases, which used to generate ontology. Thus, they are based on a set of rules and definitions for mapping relational databases to OWL Ontologies.

In this paper, we present a semantic approach for automatic ontology learning from heterogeneous relational databases. Our approach is based on Wordnet for measuring the semantic similarity between components. We apply our proposal to the alimentation risks field in order to integrate easily a set of heterogeneous sources.

3 Semantic Approach for Automatic Ontology Learning

Our purpose is to automatically generate ontology from heterogeneous relational databases using a new method, which based on semantic similarity metric and a Wordnet as a lexical database aided to select the best terms for representing ontology components. We are interested to transform the logical model of database expressed by SQL language to the class hierarchy of ontology, which presents by OWL language. In general, the process of mapping relational databases into OWL structure produces the problem of incompatible between schemata. In this context, we attempt to reduce the gap between logic and ontological models. Our semantic approach provides to generate an ontology from many relational databases in the same domain, especially the alimentation risks field. The recent development of alimentation risks analysis have led to the need for strong health information systems that provide a unified access to various scientific bases, with the aim of discovery, knowing and predicting possible threats to public health [6]. These scientific bases are often heterogeneous databases, which makes their access a complex task [12]. The proposal approach for ontology learning is of type semantic because is based on the use of two important notions related to the semantic are the Wordnet and the similarity measure. The Wordnet is one of the most widely used lexical databases for English [17]. It has been extensively used to improve the quality of data sources with its semantic relations of terms.

Otherwise, the semantic similarity measure is a central issue in different domains of computer science such as natural language processing, information retrieval, word

sense disambiguation, text segmentation, question answering and so on [16]. In our work, we focus on one popular similarity measure, Wu and Palmer's similarity (wup) between concepts (C1, C2), which is defined by the function of their distance and the lowest common subsume C to C1 and C2. Thus, it is based on the use of depth (the number of arcs) according to the following formula [22]:

$$SIMwup(C1, C2) = \frac{2 \times \text{depth(C)}}{\text{depth}(C1) + \text{depth}(C2)} \quad (1)$$

Wup similarity measure has the advantage of having good performance and the fast execution time rather than the others [1]. The full process of the proposed approach for automatic ontology learning from heterogeneous relational databases is depicted in figure 1.

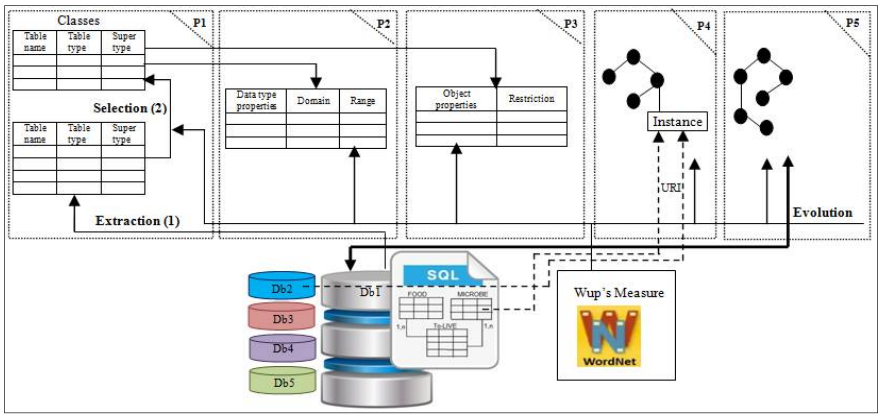

Fig. 1. A full process for automatic ontology learning

The full process for automatic ontology learning is based on four sequential processes (P1 until P4 from Fig.1) with an additional process for ensuring the evolution of obtained ontology (P5 from Fig.1). The system starts with the first process called generate classes process in order to extract and to select classes from tables' names of all databases. From the result of this process, we can extract the datatype properties of each class in the generate datatype properties process. The third process aims to generate object properties between obtained classes, and the fourth process allows integrating the records of databases as well as ontology individuals.

Regarding the fifth process, it is based on the algorithm for detecting all modifications of databases, which must be applied to their ontology. All of these five processes need to three essential elements, which are data source (databases), WordNet (a lexical database) and the Wup's similarity measure. In the following sub sections, we are going to present the detail of these processes for automatic ontology learning.

3.1 Generate Classes Process

The first process attempts to generate ontology classes from databases tables using wup's similarity based on Wordnet. Each relational database is created separately, which is heterogeneous according to the different criteria such as structure, semantic, language, format, etc. In this work, we are interested to the two first types of heterogeneity. The structural heterogeneity is defined by the different representations of data but the semantic heterogeneity corresponds to the different definitions to describe the same data [1]. We use five heterogeneous databases in alimentation risks field, which can contain the similar tables, most of the same attributes and records.

It is clear that the input of the Generate classes process is the five relational databases and the output is a set of ontology classes. The first step of this process consists to extract the tables' information of all relational databases and stored it into a new relational table called Class. The latter contains the database's number, the table's name, type using for creating tables, and the super type for representing the inheritance relation between types (in SQL is presented by the clause Under). The following table presents a part of the class table.

Table 1. The result of extracting tables' information of databases

Num_db	Table_Name	Table_type	Super_type
01	Food	Food_t	--
01	Microbe	Microbe_t	--
01	To_live	--	--
02	Alimentation	Alimentation_t	--
02	Fruit	Fruit_t	Alimentation_t
02	Vegetable	Vegetable_t	Alimentation_t
03	Microorganism	Microorganism_t	--
03	Mycotoxin	Mycotoxin _t	Microorganism_t
04	Feed	Feed_t	--
05	Nutriment	Nutriment_t	--
05	Virus	Virus_t	--
05	To_Exist	--	--

From the table presented above, the second step involves to identify the similar tables' names of five relational databases using Wup's measure, which detects the synonymy relation between concepts (tables' names) used by Wordnet. In the first, we are interested to select the similar super tables. The result is presented in the following table.

Table 2. The result of similarity measure between tables

Num_db1	Table_Name1	Table_Name2	Num_db2	Wup's similarity
01	Food	Alimentation	02	0.9231
	Food	Microorganism	03	0.4286
	Food	Feed	04	0.9231
	Food	Nutriment	05	0.9231
	Food	Virus	05	0.4615

	Microbe	Alimentation	02	0.3750
	Microbe	Microorganism	03	0.9412
	Microbe	Feed	04	0.3750
	Microbe	Nutriment	05	0.3750
	Microbe	Virus	05	0.8889
02	Alimentation	Microorganism	03	0.4000
	Alimentation	Feed	04	0.8571
	Alimentation	Nutriment	05	1
	Alimentation	Virus	05	0.4286
03	Microorganism	Feed	04	0.4000
	Microorganism	Nutriment	05	0.4000
	Microorganism	Virus	05	0.9412
04	Feed	Nutriment	05	0.8571
	Feed	Virus	05	0.4286

According to the results presented in the table above, we can deduct the similar tables existing in different databases, for example, the Food table form database01 is similar to Alimentation table from database02 and Feed table from database04 and also Nutriment table from database05. When a similarity between two concepts is greater than the threshold, then they are considered similar. It can be clearly seen that the Alimentation and Nutriment tables are semantically closet. According to high value of similarity measure between tables' names, we can obtain the following set of ontology classes: Nutriment and Microbe.

3.2 Generate Datatype Properties Process

In this process, we attempt to identify the datatype properties of ontology classes obtained from the previous process. All attributes of nutriment table from database05 became the datatype properties of nutriment class. Thus, we attempt to select other attributes from similar tables to nutriment table. It is about Food, Alimentation and Feed tables. The selection of additional attributes is based on a similarity measure between datatype properties of nutriment class and all attributes of similar tables. The attributes with a low similarity have been selected as new datatype properties (select the different attributes to the datatype properties). In the case of composite attributes related to the relational table, for example, the diameter of microbe is composed of length and width. The mapping of relational databases to ontology consists to consider composite attribute (for example diameter) as simple datatype properties and its components (length, width) as the sub property of corresponding datatype property.

Datatype properties which stand as a primary key in the relational model, are tagged with a Functional tag for restricting the object to take only one value for a given subject, and also, tagged with inverse-functional which restricts the subject to associate with only one object [13]. In the case of the attributes which stand as a foreign key, the mapping to ontological model is depicted in the third process (see subsection 3.3).

Furthermore, the domain and the range mappings of datatype properties are based on both the classes and the domain of corresponding attributes from relational tables. The following algorithm illustrates the domain and the range mappings.

```
Input: C: set of ontology classes, A: set of relational
attributes
Begin
For each attributes of A do
Begin
Rdfs :domain is corresponding class of C;
If  domain of attributes in A is 'Varchar' or 'Char' then
    rdfs:range rdf :resource="string"
else rdfs:range rdf :resource=the domain of attribute;
End;
End.
```

3.3 Generate Object Properties Process

After generating classes ontology and its datatype properties, it's time to identify the semantic relations between classes. That's the purpose of the third process in order to define the object properties between classes. From the subtypes of the relational model, which are presented in table 01, we can deduct the is_a relation and subclasses of the ontological model. So, the subclasses of nutriment class are the subtypes of similar relational tables, which are Fruit and Vegetable. Thus, the microbe class has Mycotoxin as a subclass. In addition, a wup's measure between subtypes of all similar relational tables is necessary for selecting the best subclasses of the ontological model.

About the one-to-many relationship from the relational model, the transformation to the ontological model consists to detect the foreign key from the logical model and generate two object properties with identifying the restriction between classes. For example, the following OWL code interprets the Category relation, which can have one or more records relating to a single record in microbe relation:

```
<owl: class rdf: ID="Microbe">
<rdfs: subclassof>
<owl:Restriction>
<owl:ValuesFrom>  <owl: class rdf: ID="Category">
        <rdfs: subclassof> <owl:Restriction>
        <owl:someValuesFrom rdf:resource="#Microbe/">
</ValuesFrom>
        <owl:onProperty> <owl: Objectproperty
rdf:ID="has-Microbe"/>  </owl:onProperty>
```

```
        </owl:Restriction> </rdfs: subclassof>
</owl: class> </ValuesFrom>
<owl:onProperty> <owl: Objectproperty rdf:ID="has-
category"/>  </owl:onProperty>
</owl:Restriction>
 </rdfs: subclassof>
</owl: class>
```

We apply the same rule for presenting many-to-many relationship. In this kind of relation, we use the tag <owl:someValuesFrom> for each restriction related to the classes.

3.4 Generate Instances Process

The final process attempts to generate instances (or individuals) of ontology classes corresponds to an appropriate record from relational tables. Thus, the same instance can be related to many relational tables. In our method, we create a new special attribute in the database that represents a pointer referring to an instance of ontology class [1]. So, each instance is related to its records via this special attribute, which contains the URI (Uniform Resource Identifier) of the corresponding instance from the ontology. An instance of ontology class is a set of values grant to each datatype property. So, the superfluous attributes from relational table have been eliminated. Form the special attribute; we can manage consistency between the content of the database and obtained ontology (more detail in subsection3.5). From these four processes, the ontology has been created from relational databases.

3.5 Ontology Evolution Process

The ontology evolution process allows the obtained ontology to change according to its relational databases [2]. Each modification of relational databases content involves the same operation in the generated ontology. It is through the special attribute, which contains the URI of the instance of ontology class, we can ensure the auto evolution of ontology. The following algorithm illustrates the evolution of ontology by the evolution of its database.

```
Input : DB : database, Ont: ontology
           Op : {‘any’, ‘insert’, ‘delete’, ‘update’}
Begin
For  each i^th table of DB Do
 Begin
   If op= ‘any’ Then Exit
      ELse if  op= ‘Insert’ Then
```

```
       Begin
              Insert into Ont the new instance according
to the new record;
              Update the SA a special attribute by the
URI of the new instance ;
         End if
   Else if op= ‘delete’ Then
       Begin
             URI := SA[k]    //extract the uri of Kth ele-
ment to remove
             Delete the instance from Ont;
             Delete SA[k];
         End if
     Else //update
       Update the kth instance of Ont;
  End for;
End.
```

The ontology evolution algorithm presented above takes as input, the database, the ontology and a set of operations, which can apply to both database and ontology. There are three principal operations: insert, delete and update. If any modification applied to the database (operation=any) then exit the program else execute the ontology evolution. The main idea in this algorithm is to use the special attribute SA for facilitating the reference of the instance to delete or to update [2]. In the insert operation, the first step consists to insert a new instance according to the new record. The second step allows giving the URI of this new instance and updates the SA value by this URI without changing of ontology. Therefore, the algorithm affects the ontology evolution according to the evolution of its relational databases.

4 Experimentation

In this section, we provide an experimentation of our approach. To do that, we have implemented the above processes into our prototype implementation of automatic ontology generation that based on Java as a programming language. The prototype provides an intuitive interface to represent the results of different processes of ontology learning. The following figure shows the execution of our prototype.

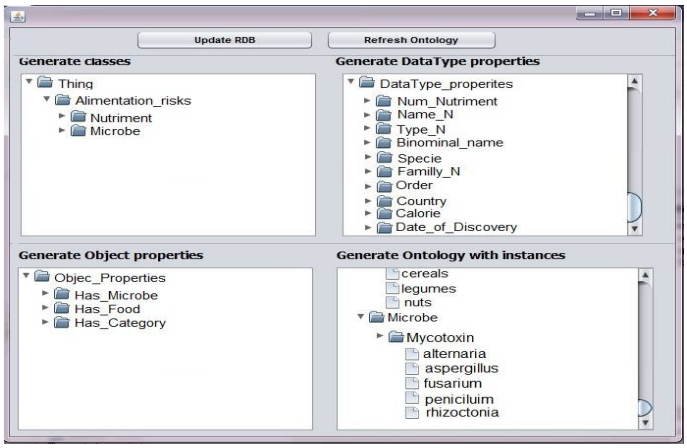

Fig. 2. A screenshot of generation ontology prototype

The interface of our prototype is split into four parts (01, 02, 03, 04) which represent results of four processes (generate classes, generate datatype properties, generate object properties and generate instances). Part 04 shows the generated ontology with its instances. The interface also offers the possibility to update relational databases using the button 'update RDB'. After updating the database, we can refresh the generated ontology by 'refresh ontology' button and show the ontology evolution in part 04.

Otherwise, in order to identify the advantage of our prototype compared to other similar tools, we compare it to three popular tools which are DataMaster [19], KAON2 [11] and RDBToOnto [12]. Hence, we use the same relational database among the five databases in alimentation risks field that we created for validating our approach. We are selected three comparative criteria: the functionality of a tool, which define the principal function of the tool, the features, which are a set of characteristics such as appearance of technical errors, the completeness ontology, the cost and time for ontology learning, etc. The third criterion is the size of a tool, which represent a disk space required by a tool setup. The following table shows the result.

Table 3. Comparison between our prototype and other similar tools

Tool	Functionality	Features	Size
Our prototype	- Ontology generation from many RDBs. - Ontology evolution.	- Complete ontology. - Nothing errors. - Quick building. - Allows updating database. - Evolutionary ontology.	209 ko
DataMaster	- Ontology generation from one RDB.	- Incomplete ontology (some relationships do not detected). - RDB constraints are not treated.	775 ko
KAON2	- Manipulating OWL- DL ontologies.	- Ontology manually generated which its instances have been ex-	2305 ko

	- Extracting ontology instances from RDB.	tracted from RDB.	
RDBToOnto	- Ontology generation from RDB.	- Less complete extraction rules.	13119 ko

From the table, it can be seen that our approach for automatic ontology learning is more efficient and return a complete ontology for generating of essential components of an ontology. Thus, our prototype does not produce any technical errors during the ontology learning. So, it facilitates a quick building and reduces the cost and time for ontology learning. This prototype allows the ontology evolution as evolution evolves of its relational databases. In addition, we can see that our prototype having the smallest disk space compared to DataMaster, KAON2, and RDBToOnto. Unlike KAON2, is the framework for building ontology without instances; it contains a module for extracting ontology instances from relational databases. For future work, another criterion we will take into accounts such as the time of ontology generation, the performance, and the complexity.

5 Conclusion

We have presented a new approach for automatic ontology learning and evolution from relational databases RDB. This approach is based on five processes: Generate classes process which builds classes from relational tables, Generate datatype properties process which generates properties of the classes obtained from the first process, Generate object properties process which uses the relations between tables in order to create relationships between classes, Generate instances process which considers the data of RDB as instances of ontology, and Ontology evolution process which modifies the ontology as the modification of its RDB. We use in this approach a Wup's measure, which detects the synonymy relation between concepts used by Wordnet. Thus, the proposed approach has been validated by the development of the prototype to show the effectiveness and automation ontology learning. This prototype has been tested and evaluated by a comparison study with similar tools of ontology learning from databases and the result shows the novelty of our contribution.

Future work includes depth analysis of the performance of our prototype using big databases. The results of this work will help to solve several semantic problems such as semantic integration, semantic information retrieval, and semantic querying, etc.

References

1. Aicha, Aggoune., Abdelkrim, Bouramoul., Mohammed, K, Kholladi.: Mediation system for dealing with semantic problems in databases. International Journal of Data Mining, Modelling and Management 9(2), 99–121 (2017).
2. Aicha, Aggoune.: Traitement de l'Hétérogénéité Sémantique pour l'Exploration des Sources de Données Multimédias. Phd's thesis. Constantine 2 university, Algeria (2017).
3. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific american 284(5), 28-37 (2001).

4. Bumans, G.: Mapping between Relational Databases and OWL Ontologies: an example. Scientific Papers, University of Latvia, 756, 99-117 (2010).
5. Dadjoo, M., Kheirkhah, E. : An approach for transforming of relational databases to OWL ontology. Int. J. Web & Semantic Technology, (IJWesT) 6(1), 19-28 (2015).
6. De, Valk, H., Salvat, G. : Alimentation et risques infectieux : enjeux et stratégies pour limiter l'impact sur la santé. Les Tribunes de la santé, 49(4), 61-68 (2015).
7. Dogdu, E., Ozbayoglu, A,M., Benli, O., Akinc, H,E., Erol, E., Atasoy, Ercin, O.: Ontology-centric data modelling and decision support in smart grid applications a distribution service operator perspective. In: International Conference on Intelligent Energy and Power Systems (IEPS), pp. 198-204. IEEE, Ukraine (2014).
8. Drame, k.: Contribution à la construction d'ontologies et à la recherche d'information: application au domaine médical. Phd's thesis. University of Bordeaux, French (2014)
9. Fahad, M.: Er2owl: Generating owl ontology from ER diagram. In: International Conference on Intelligent Information Processing, pp. 28-37. Springer, China (2008).
10. Ho, L,T., Tran, C,P., Hoang, Q.: An Approach of Transforming Ontologies into Relational Databases. In: Asian Conference on Intelligent Information and Database Systems, pp. 149-158. Springer, Cham (2015).
11. http://kaon2.semanticweb.org/, last accessed 2017/11/15
12. http://www.taoproject.eu/researchanddevelopment/demosanddownloads/RDBToOnto.html
13. http://www.w3.org/TR/2004/REC-owl-guide-20040210/, last accessed 2017/11/14
14. Krivine, S., Nobécourt, J., Soualmia, L., Cerbah, F., Duclos, C. : Construction automatique d'ontologie à partir de bases de données relationnelles: application au médicament dans le domaine de la pharmacovigilance. In: Actes des 20es Journées Francophones d'Ingénierie des Connaissances, IC, pp. 1-12. Tunisia (2009).
15. Kumova, B, I.: Generating ontologies from relational data with fuzzy-syllogistic reasoning. In: International Conference: Beyond Databases, Architectures and Structures, pp. 21-32. Springer, Cham (2015).
16. Li, M., Du, X, Y., Wang, S.: Learning ontology from relational database. In: International Conference Machine Learning and Cybernetics, pp. 3410-3415. Vol 6, IEEE, China (2005).
17. Miller. G, A.: WordNet: a lexical database for English. Communications of the ACM, 38(11), 39-41 (1995).
18. Nakhla, Z., Nouira, K.: Automatic approach to enrich databases using ontology: Application in medical domain. Procedia Computer Science, 112, 387-396 (2017).
19. Nyulas, C., O'connor, M., Tu, S.: DataMaster - a Plug-in for Importing Schemas and Data from Relational Databases into Protégé. In: 10th International Protégé Conference, pp. 1–3. Hungary (2007).
20. Tahat, S., Ahmad, K.: A method on lexical disambiguation in distributed heterogeneous autonomous database. In: International Conference on Research and Innovation in Information Systems, ICRIIS, pp. 330-335. IEEE, Malaysia (2013).
21. Upadhyaya, S., Kumar, P.: ERONTO: A Tool for Extracting Ontologies from Extended E/R Diagrams. In: Proceedings of SAC"05 ACM Symposium on Applied Computing, pp. 666-670. Santa Fe, USA (2005).
22. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: ACL 1994: Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, pp. 133–138. USA (1994).
23. Cerbah, F.: Learning Highly Structured Semantic Repositories from Relational Databases: In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds) The Semantic Web: Research and Applications. ESWC 2008. Lecture Notes in Computer Science, vol 5021, pp.777-781. Springer, Berlin, Heidelberg (2008).