Indian Journal of Science and Technology Vol 8(17), DOI: 10.17485/ijst/2015/v8i17/62219, August 2015
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Hybrid Layered Approach for Ontology Matching

V. Viniba* and N. Sairam

School of Computing, SASTRA University, Thanjavur - 613401, Tamil Nadu, India; viniba1992@gmail.com

Abstract

The technique of ontology matching acts as a prerequisite for performing various activities in the semantic web. Most Ontology matching methods compare class names only based on their similarity irrespective of their real meaning; failed to specify the type of semantic relation. The proposed hybrid layered approach calculates syntactic, semantic and structural similarity between the classes from input ontologies in successive layers. Finally, the alignment layer generates the final matching results by combining the results obtained from the previous layers and generates semantic mappings between them. This approach overcomes the risk of generating incorrect matching results. The test dataset from OAEI are used for evaluating the proposed hybrid approach. The results obtained proved that the proposed approach overtakes the other existing methods by generating correct matching results thereby improving the accuracy of the results achieved. This approach can also be applied to match other entities of the ontology. Since the layers provide matching results based on user – defined threshold, it has the advantage of excluding the incorrect mappings. Thus the proposed hybrid layered approach helps to learn the knowledge hidden between the local ontologies leading to the improved precision.

Keywords: Linguistic Features, Ontology Classes, Ontology Matching, Semantic Relations, Synset

1. Introduction

The technique of ontology mapping is realized as a key provider for integrating the semantically related information available from different data sources[1–3]. Ontology mapping is categorized into different types based on their roles[4]. Most of the research work is based on developing a mapping system that can able to generate mappings between the native ontologies from similar or corresponding domains[5]. Ontology mapping depends on numerous semantic technologies to solve heterogeneity problem for information integration. Ontology matching is one among the key technology that aims to find the semantically interrelated entities from different ontological domain[6].

Similarity among the entities of the ontology can be computed by two techniques. They are single strategy and joining multiple strategies. Single strategy combines all the features of the ontology entities and express them as a single function, whereas the multiple strategy approach express multiple similarity function based on various features of the ontology entities[7–9]. Matching systems such as AgreementMaker[10], COMA[11] relies on user feedback to prefer the best matching schemes as they offer more than one similarity methods to measure the similarity between the entities. Currently, the combination technique grows into a popular method, owing to its comfort of extension and elasticity. It is evident from frameworks such as RiMOM[8], ASMOV[12], that combining the results of the multiple matching schemes can expand the total matching results.

Class name similarity computation between input ontologies is the primary step performed in every ontology matching process. Class names have a similar label with diverse meanings and different labels with related meanings. Most of the traditional similarity algorithms matching calculate only the syntactic similarity between the class names irrespective of their real meanings. This automatically affects the ratio of total matching results and produces incorrect mappings between the ontologies. Matching tools such as COMA++[13], Falcon[14] are used in finding only the accurate or correspondent matching pair of classes in the input ontologies, irrespective of their inherent relationship present between them. This may

*Author for correspondence

result in restriction in studying the type of information present between the local mappings.

A Hybrid layered approach is proposed in this paper. This approach computes similarity between the class names of the source and the target ontologies using various features attached to the ontology classes. The proposed ontology matching approach is performed in four layers. The first layer calculates the syntactic similarity between the entities of the source and the target ontology. Second layer computes the semantic similarity in two steps. The third layer finds some additional matching entities by computing structural similarity between the ontology classes. The final layer combines the match results obtained from the previous layers and remove the redundant and inconsistent matching classes. Then, the type of semantic relation existing between the matched correspondences is specified and final mappings between the input ontologies are generated. The proposed system has the advantage of generating precise mappings with improved accuracy.

Some of the tools that depend on multiple matching methods to perform ontology matching are discussed below.

Falcon–AO[14] is an automated ontology matching system and is provided with a pool of similarity matchers where the central organizer is responsible for executing the matchers and combining the match results obtained from various matching strategies. A user interface was implemented in this system in order to make the generated alignments easily accessible to the users. Threshold values are chosen based on the obtained match correspondences to the number of entities in the ontology. The main weakness of this system is that it matches the entities with high string similarity without considering the structures of the given ontologies. It is restricted to generate mappings with their semantic relation.

Agreement Maker[10] is reflected as one of the attractive and advanced ontology matching system which is provided with a convenient user interface for setting various parameters. It is highly customized with a flexible architecture that allows numerous matching techniques to be executed on the input entities with various features of the ontology. The decision on which the format of the generated mappings should be is driven either by the user intervention or automatically.

ASMOV[12] (Automated Semantic Mapping of Ontologies with Validation) is an innovative algorithm that is developed upon the concepts used by former systems. It presents an algorithm that programs the ontology matching process equipped with option of user interaction. It automatically assigns and adjusts weights based on the presence of features within the ontologies. This system also comprises a stage of semantic validation to verify the generated alignments in order to eliminate the semantic inconsistencies. This system is recommended to assign similar weights for all the similarity measures used and the performance of the vocabulary reference system (WordNet) used is slower.

COMA++[13] is a generic framework which matches the components of the ontology by decomposing the entire ontology into smaller fragments. This uses entity and structural matchers which can be nominated and arranged based on the user needs. It has a mapping pool which act as a repository for the generated mappings and it can be reused in various applications. An effective user interface is delivered with this system to configure the matcher selection and to combine the matching strategies. But, this system is restricted to express the type of semantic relations existing between the derived mappings.

S–Match[15] consists of several components for the transformation of data structures into ontologies and finds the semantic similarity between them by using WordNet. It is implemented with three different types of semantic matching procedures, each meets different requirement. It is highly flexible that it allows to integrate the new techniques as pluggable components. Research work has been done on this system to support and enrich the output alignment formats

RiMOM[8] is an optimized matching system which relies on multi–matching technique to compute similarity between input ontologies. Firstly, the similarity values between two inputs are computed using different matching methods independently. Then, the initial matching results are extracted from different matching methods independently by applying threshold value filters. The initial matching results are combined based on some global threshold. It applies a strategy selection method in order to elect the best matching method. The process is repeated iteratively until no new mappings are found. It is assisted with an optional user interface in order to get newly found mappings or to correct the inaccurate mappings. The error analysis of RiMOM reports that it fails in finding the precise mapping expression. RiMOM challenges to provide a typical language for delivering the mappings for client applications.

2. Detailed System Architecture

This framework works on a consistent method for matching the entities of input ontologies where the final mappings obtained are consistent in the manner. The overall architecture of the proposed systemis shown in Figure 1.

3. Description of the Proposed Approach

The proposed system starts the matching process by loading source and target ontologies and extracts the valuable entities such as ontology classes, properties and their corresponding structural relations from them. If there is no existence of ontology in OWL language we can make use of the algorithm[16] to convert the database schemas into OWL ontology.

Class is the core feature of ontology and they are responsible for organizing the hierarchical structure for the ease of application processing needs. Discovering the similarity between the classes of the given source and target ontology is the primary step in every ontology matching process. Retrieval of correct match results from the primary process may direct the appropriate path for matching the other relative entities of the ontology. It is therefore proposed to execute the process of concept name matching in several layers of refinement in order to achieve the correct match results. This system performs syntactic, semantic and structural matching in successive layers between the classes of the given ontologies to produce the initial match results. The results obtained from the three layers of the system are aggregated to produce the final mappings in the alignment layer.

3.1 Syntactic Layer

Generally in ontologies, the class names are usually enclosed in rdfs:label tags. A label could be represented as a string of characters without any delimiters which could be a term or else a group of words. They are used to afford a human–readable explanation of the classes and are distinctive in ontologies. The syntactic layer [L_{syn}] begins the matching process by calculating string matching between the label of the classes of the input ontologies.

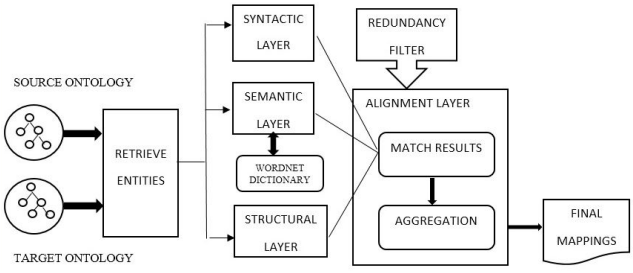

Figure 1. Overall Architecture of the proposed system.

3.1.1 *Edit Distance based Similarity*

String similarity can be computed[17,18] usually by taking as input the label of two classes, then calculate approximately the distance between them by using some arithmetic distance operations that returns a tangible number to signify the similarity. As a result, the output will be a number n ∈ [0, 1] to signify the confidence of their match.

This system computes the string similarity between the class names of the input ontologies by using the edit distance similarity metric. The edit distance between the two strings is defined as the least number of operations required to transform the source string into the target string. The operations that can be performed are addition, replacement and omission of characters. Every class name in source ontology is compared with every class name in the target ontology. The distance values obtained from this metric are arranged in a $|E_s|.|E_t|$ similarity matrix S_1 where $E_s=\{s_{c1},s_{c2}\ldots\ldots s_{cn}\}$ and $E_t=\{t_{c1,}t_{c2}\ldots\ldots t_{cm}\}$. E_s and E_t contains the set of class names extracted from the source and target ontology. Each class pair in the matrix is provided with their distance value. The interested class pairs are extracted by selecting the class pair with values above the defined threshold and added to the result set which is represented by $RL_{syn} = \{s_{c1}=t_{c2},s_{c4}=t_{c6},\ldots..,s_{cn}=t_{cm}\}$. The threshold T is chosen either based on user feedback or by put on k–means clustering algorithm to the matrix values. The similarity matrix obtained for the sample inputontologies is shown in Table 1.

Table 1. Similarity matrix

BibTeX/Myonto	Published	Booklet	Thesis	Manual	Entry
Book	0.77	0.42	0.83	0.83	0.8
Booklet	0.66	0	1	0.85	1
Masterthesis	0.83	0.91	0.58	0.91	0.91
Entry	0.88	0.85	0.83	0.66	0
Unpublished	0.18	0.81	0.81	0.72	0.90

3.2 Semantic Layer

Linguistic features remain essential for developing a primary set of alignments which can be refined by using other types of matching. Even though, the string similarity measures are used to provide most essential clues to check whether two class terms are identical or not; it is important to discover for semantic relations between them based on various descriptions attached to them. The semantic layer checks for similarity between the classes in two steps. First, the document–based similarity is carried out, followed by using a third party knowledge source such as WordNet.

3.2.1 Document based Similarity

The next process involved in this layered approach is the computation of similarity between the classes of the given ontologies based on the different features attached to the classes. The features of the ontology entities are defined as the various descriptions devoted to each entity in the ontology such as labels, comments, etc. The feature description of the entities is collected and transformed into virtual documents and the document based similarity is computed between them. The document based similarity is calculated by using the vector based similarity metric.

Labels are one of the annotation properties that are used to offer a human–understandable form of an ontology entity. It may be a short word or a long text. Comments are also one among the annotation properties that offers human–understandable explanation of an ontological entity. Since comments are descriptions usually consist of one or more sentences. If there is any other type of annotation properties they are also considered as features of the ontology entity. Thus the overall feature description of an entity E is defined by

D (E) = β1 ∗ group of words found in the local name of E +

β2 ∗ group of words found in the rdfs: label value of E +

β3 ∗ group of words found in the rdfs: comment value of E +

β4 ∗ group of words found in other annotation value of E.

Where β1, β2, β3 and β4 are static, rational numbers between 0 and 1, which specify the weights for each kind of description.

The virtual documents are then created by collecting the description D (E) of the class entity with the help of the method applied in previous system[19]. Now the similarity between the virtual documents is computed by means of vector based similarity metric. The virtual documents are then denoted as a vector where the components of the vector are nothing but the numerical weights assigned to the words found in the document. The weight of a word is computed mainly in order to reflect its importance in the virtual document. The weight of a word x in aclass entity's virtual document y is calculated using TF–IDF method[20]. Thus,virtual documents are created for every class entity in the source and thetarget ontology.

The vector based similarity metric estimate the cosine similarity between the source and target class entity's virtual document. The similarity values obtained are organized in a similarity matrix S_{ij} where the value present in the i^{th} row and j^{th} column denote the cosine similarity between the i^{th} classin E_s and j^{th} class in E_t where $E_s = \{s_{c1}, s_{c2} \ldots\ldots s_{cn}\}$ and $E_t = \{ t_{c1}, t_{c2} \ldots\ldots t_{cm}\}$. Each numerical value in the matrix S_{ij} denotes the document similarity between the class pairs. Again, a threshold value e should be chosen where the option is left to user choice or guided by k-means clustering algorithm. The similarity values which exceed the given threshold are extracted and added to the result set RL_{doc}.

3.2.2 Background Knowledge based Matching

After matching the classes of the input ontologies lexically it is necessary to execute matching based on some background knowledge sources. Since the entities in different ontologies are expressed using different terms, the matching between the entities that are semantically related cannot be found. This problem can be solved by matching two ontologies using a third widespread background knowledge sources such as WordNet,which aids in finding the semantic similarity between the entities even if those entities are lexically or structurally not overlapped. WordNet is the most widespread lexicon in English language where the words are organized semantically. WordNet contains set of synonyms for each word known as synset. Synsets offer diverse semantic associations which are shown in Table 2.

If one of the synset of the source word is same as that of the target word, then they are considered to be semantically equal. It is seen that similarity has an extensive sense than synonymy. Here two words are said to be semantically similar if they have a common synset (synonymy). In our sample input ontologies the class terms are initialized in set E_s and E_t. For instance the class term **book** in E_s is semantically equal to class term **record** from E_t. The synset of Book is

Table 2. Types of Semantic Relation

S.No.	Semantic Relation	Meaning	Relation type
1	Synonyms	Similar/equivalent	Equal, same as
2	Antonyms	Opposite/Disjoint	Mismatch
3	Hypernyms	Superset of	Is-a
4	Hyponyms	Subset of	Inverse is-a
5	Meronyms	Part-of	Belongs to
6	Holonyms	Has-a	Has-a

Book→[Book,volume],[record,record,book,book],[script,book,playscript],[ledger,leger,account book,book of account, book].

Since the term record is present in the synset of book, they are semantically equal even though they are syntactically different. If the synset of the source class term contains an antonym for the synset of the target class term, then they are considered to be disjoint in relation. If the synonymy and hypernymy relations are not discovered then it is recommended to check for Hypernyms–hyponyms relations. Given a source class C1 and a target class C2 we detect for is a type of semantic relation between them using WordNet(see Algorithm 1).

Algorithm 1. Hypernym– based matching algorithm.

Input:

Source Class set $E_s=\{s_{c1},s_{c2}\ldots\ldots s_{cn}\}$,

Target Class set $E_t=\{t_{c1,}t_{c2}\ldots\ldots t_{cm}\}$,

WordNet Dictionary W.

Output:

Result set $RL_W$with selected matched classes

1: Let $RL_W \leftarrow \emptyset$;
2: i ← 1 and j ← 1;
3: **for** all class s_{ci} in E_s do
4: **for** all class t_{cj} in E_t do
5: **if** W contains s_{ci} and $t_{cj,}$ then
6: get Hypernyms H of s_{ci} and t_{cj}
7: **if** H(s_{ci}) contains t_{cj} then
8: $RL_W \leftarrow RL_W$ U (s_{ci} , t_{cj}) ;
9: Map s_{ci} is-a t_{cj};
10: i = i + 1;
11: j = j + 1;
12: **end if**;
13: **end if**;
14: **end for**;
15: **end for**;
16: return RL_W.

The Hypernym based matching algorithm match the classes between the source and target ontology for is a type of semantic relation. WordNet background method make full use of and derive benefit from linguistic relations such as synonyms, hypernyms and hyponyms in order to enhance its usefulness.

3.3 Structural Layer

Next, the structural layer relates the two classes from input ontologies with respect to the relations of these classes with other classes in the ontology. It is based on the assumption that the more similar the two classes are, the more similarity will exist between their relative entities. The match results generated in this layer are obtained by calculating similarity of the source class with that of the relative features present under the target class. Such relative features that exist under each class are subclasses and super–classes. The layer considers that if two classes lie in identical places with a type of semantic relation (is–an otherwise part–of) which are in relation to classes already associated with the input ontologies, and then they are expected to be comparable as well.

For instance, If two classes s_c and t_c that are already aligned (s_c,t_c), then the structural layers extends with a similarity value for any pair of classes (s'_c,t'_c), such that they are in a type of semantic relationship with the classes s_c and t_c respectively. This layer follows some rule in order to obtain the accurate matching. If the super–class or sub–class of two classes is identical, then they are considered to be same. The key concept of this matching is that two classes are considered to be similar if they have their similar relative entities and same attributes.

3.4 Alignment Layer

The results obtained from syntactic, semantic and structural layer are aggregated in this layer to generate the final matching results. Thus, for each pair of classes in input ontologies, there will be several similarity values. Therefore, the main work of this layer is to combine the results obtained from the previous layer in a most effective way in order to get efficient results. The result sets RL_{syn}, RL_{doc}, RL_W and RL_{str} are combined to generate the output result set R. Since the matching pairs present in the result set of previous layers are refined by threshold, this layer does not need to perform threshold based extraction. It is enough to remove the redundant matching pairs from the output result set R. The matching class pair from the

result set R is mapped with the type of semantic relation to produce the final mappings between the source and target ontology.

4. Implementation and Result Evaluation

This hybrid matching system is implemented on the Java platform. This approach has used OWL API[21] to load and parse the ontologies available in OWL and RDFS formats. For the semantic similarity, this system interacts with JWordNet, a WordNet library supported by Java to query for various senses of a word. Precision, Recall and F1–measure are the metrics used to measure the performance of the hybrid approach.

Precision is the proportion of the intersection of entities matched automatically and entities matched manually to the automatically matched entities.

Recall is the proportion of the intersection of entities matched automatically and entities matched manually to the manually matched entities.

The performance result of the proposed system is compared to other matching system (Falcon–AO,COMA++, RiMOM) discussed in the related work. The precision and recall value obtained from the proposed approach has advantages over the other matching systems and is merely equal to that of the RiMOM. The graph comparison of performance metrics obtained through the hybrid system with other matching systems is shown in Figure 3. This is because of the proposal of the matching process using

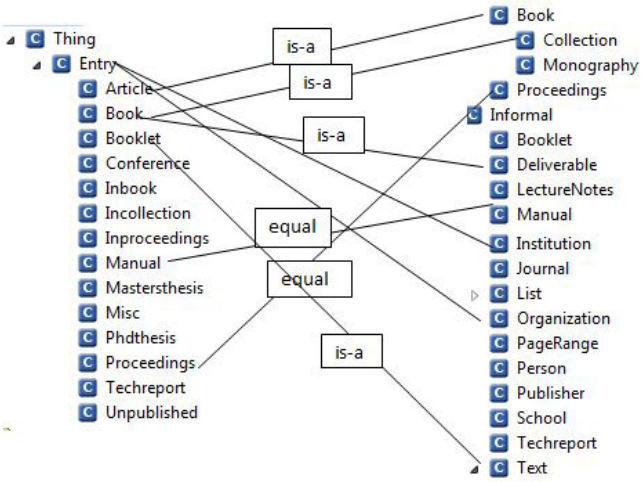

Figure 2. Final Mappings achieved for sample input ontologies.

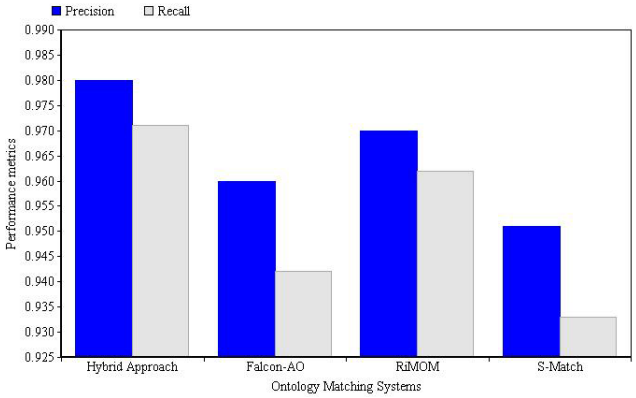

Figure 3. Comparison of the Hybrid approach with other systems.

different similarity metrics and complete influence of the semantic information present over the ontologies.

The experimental dataset taken to evaluate the proposed approach is the BibTeX dataset. These ontologies are one among the Ontology Alignment Evaluation Initiative (OAEI) datasets and are accessible from the web[22]. There are two ontologies in BibTeX datasets namely edu.mit.visus.bibtex.owl and myOnto.owl. The Former isa moderately accurate copy of BibTeX presented as ontology and latter one is the extended version of the former with some additional concepts. The bibtex ontology consists of 15 class entities and myonto ontology consists of 39 class entities. The mapped concepts between the source and the target ontology obtained using the proposed approach is shown in Figure 2. This Hybrid system attains a high precision of 0.98 and recall value of 0.97 and the final results are mapped with their type of semantic relation.

5. Conclusion

A novel ontology alignment methodology is proposed in this paper. This is a hybrid approach which accomplishes the matching process in several layers and merges the results obtained from those layers. The proposed approach compares the class terms of the source and the target ontology by using syntactic, semantic, structural matching techniques. Finally, the alignment layer combines the match results obtained from the previous layer and define them with the appropriate type of semantic relation. It follows the same technique for matching the other entities of the ontologies. Since the layers provide matching results based on user defined threshold, it has

the advantage of excluding the incorrect mappings. Also, it helps to learn the information hidden between the local source and the target ontologies. The results obtained are more precise leading to the overall improvement in the accuracy. In the future, it is planned to extend the hybrid system in two ways. First one is to integrate additional matching methods and second one is to work with large–scale ontologies.

6. References

1. Hassanzadeh O, Lim L, Kementsietsidis A,Wang M. A declarative framework for semantic link discovery over relational data. Proceedings of the 18th International Conference on World Wide Web (WWW '09); p. 1101–2.
2. Abirami AM, Askarunisa A. A semantic based approach for knowledge discovery and acquisition from multipleweb pages using ontologies. International Journal of Weband Semantic Technology. 2013 Jul; 4(3):193–200.
3. Karthikeyan K, Karthikeyani V. Ontology based concept hierarchy extraction of web data. Indian Journal of Science and Technology. 2015; 8(6):536–47.
4. Kalfoglou Y, Schorlemmer M. Ontology mapping: the state of the art. The Knowledge Engineering Review. 2003; 18:1–31.
5. Pavel S, Euzenat J. Ontology matching: state of the art and future challenges. IEEE Transactions on Knowledge and Data Engineering. 2011; 25(1):158–76.
6. Doan A, Madhavan J, Dhamankar R, Domingos P, Halevy A. Learning to match ontologies on the semantic web. The VLDB Journal. 2003; 12:303–19.
7. Choi N, Song IY, Han H. A survey on ontology mapping. ACM SIGMOD Record. 2006; 35:34–41.
8. Li J, Tang J, Li Y, Luo Q. Rimom: a dynamic multistrategy ontology alignment framework. IEEE Transactions on Knowledge and Data Engineering. 2009; 21:1218–32.
9. Wang Z, Li J, Zhao Y, Setchi R, Tang J. A unified approach to matching semantic data on the Web. Knowledge–Based Systems. 2013; 39:173–84.
10. Cruz IF, Antonelli FP, Stroe C. AgreementMaker: Efficient matching for large real–world schemas and ontologies. Proceedings of VLDB Endow. 2009 Aug; 2:1586–9.
11. Do HH, Rahm E. Coma: a system for flexible combination of schema matching approaches. Proceedings of the 28th International Conference on Very Large Data Bases (VLDB '02); p. 610–21.
12. Jean–Mary YR, Shironoshita PE, Kabuka MR. Ontology matching with semantic verification. Journal of Web Semantics. 2009;7(3):235–51.
13. Aumueller D, Do S, Massmann HH, Rahm E. Schema and ontologymatching with coma++. SIGMOD; p. 2005.
14. Hu W, Qu Y. Falcon–AO: a practical ontology matching system. Web Semantics: Science, Services and Agents on the World Wide Web 6; 2008. p. 237–9.
15. Giunchiglia F, Shvaiko P, Yatskevich M. S–Match: An algorithm and implementation of semantic matching. Proceedings of the European Semantic Web Symposium LNCS. 2004; 3053:61–75.
16. Alalwan N, Zedan H, Siewe F. Generating OWL Ontology for Database Intgeration. Proceedings of Third International Conference on Advance in Semantic Processing; 2009. p. 22–31.
17. Bunke H, Csirik J. Parametric String Edit Distance and Its Application to Pattern Recognition. IEEE Transactions on Systems, Man, and Cybernetics. 1995; 25:202–6.
18. Cohen WW, Ravikumar P, Fienberg SE. A Comparison of String Distance Metrics for Name–Matching Tasks. Proceedings of 2nd Web; 2003. p. 73–8.
19. Qu Y, Hu W, Cheng G. Constructing Virtual Documents for Ontology Matching. Wide Web 4; 2006. p. 243–62.
20. Salton G, Yang CS. On the specification of term values in automatic indexing. Journal of Documentation. 1973; 29:351–72.
21. Available from: http://owlapi.sourceforge.net/ .15/4/2014
22. Available from: http://oaei.ontologymatching.org/2008/ .15/4/2014