2008

# Ontology matching using vector space

Zahra Eidoon
*University of Tehran, Iran*

Nasser Yazdani
*University of Tehran*

Farhad Oroumchian
*University of Wollongong in Dubai*, farhado@uow.edu.au

## Publication Details
Eidoon, Z., Yazdani, N. & Oroumchian, F. 2008, 'Ontology matching using vector space', in European Conference on Information Retrieval, 30 Mar - 3 Apr, Glasgow, Scotland, Lecture Notes in Computer Science, vol. 2008, no. 4956, pp. 472-481.

# Ontology Matching Using Vector Space

Zahra Eidoon [1], Nasser Yazdani [1] and Farhad Oroumchian [2]

[1] ECE Department, University of Tehran, Tehran, Iran
[2] University of Wollongong , Dubai;Control and Intelligent Processing Center of Excellence, Faculty of Eng., Univ. of Tehran,Tehran,Iran
{z.eidoon , yazdani}@ece.ue.ac.ir
FarhadOroumchian@uowdubai.ac.ae

**Abstract.** Interoperability of heterogeneous systems on the Web will be achieved through an agreement between the underlying ontologies. Ontology matching is an operation that takes two ontologies and determines their semantic mapping. This paper presents a method of ontology matching which is based on modeling ontologies in a vector space and estimating their similarity degree by matching their concept vectors. The proposed method is successfully applied to the test suit of Ontology Alignment Evaluation Initiative 2005 [10] and compared to the results reported by other methods. In terms of precision and recall, the results look promising.

**Keywords:** semantic web, ontology matching, vector space.

## 1 Introduction

The current World Wide Web has over 22.47 billion pages [17], but the vast majority of them are in human readable format only. In order to allow software agents to understand and process the web information in a more intelligent way, researchers have created the Semantic Web vision [15], where data has structure. Like the Web, the semantic Web will necessarily be distributed and heterogeneous. Therefore, the integration of resources found on the semantic Web is a key issue. A standard approach to the resulting problem lies in the use of ontologies for data description. Ontologies allow users to organize information into taxonomies of concepts, each with their properties, and describe relationships between concepts [16]. However, the available ontologies could themselves introduce heterogeneity: given two ontologies, the same entity can be given different names in each of them or simply be defined in different ways, whereas both ontologies may express the same knowledge but in different languages. So, one of the key challenges of Semantic Web is to find semantic correspondences between ontologies.

The underlying problem, which we call the ontology matching (or alignment), is the operation of taking two distinct ontologies, finding a set of entities with similar relationships which exist in both ontologies and return the similar entities. Shvaiko et al. classifies ontology alignment techniques in two general categories: element-level techniques and structure-level techniques [5]. The former techniques concentrate just on individual elements while in latter approaches the structural arrangement of elements and their relation to each other is more of interest. The structural-level techniques involve Graph-based techniques which consider the input as labeled graph, Taxonomy-based techniques which consider only the specialization relation, Repository of structures which stores schemas/ontologies and their fragments together with pair-wise similarities (e.g., coefficients in the [0 1] range) between them and finally Model-based algorithms which handle the input based on its semantic interpretation (e.g., model-theoretic semantics). Furthermore, ontology matchers can be categorized into automatic and semi-automatic techniques. Automatic ontology matchers are those which perform their operation independent of human operator, while semi-automatic techniques are dependent on user preferences.

This paper presents an automatic taxonomy-based ontology alignment technique that is based on a vector matching method. Any ontology consists of a set of concepts and each concept is described by a set of properties. These concepts and properties define a space such that each distinct concept and property represents one dimension in that space. Modeling ontologies in multi-dimensional vector spaces will enable us to use vector matching methods for performing ontology alignment. An iterative approach has been employed to achieve convergence, in which vectors representing ontology concepts and properties are matched iteratively and their similarity degree is estimated. In order to model two ontologies in a vector space, RDF [1] and OWL [7] subclass predicates are utilized and concepts are described with respect to their ancestors and successors and properties. Properties are also described with respect to their domain and range concepts.

The rest of the paper is organized as follows. In section 2 we discuss state of the art of matching systems from the structured base ontology matching perspective. Our approach is presented in section 3. Experimental results are reported in Section 4. Finally, Section 5 contains some conclusions and future work.

## 2 Related Work

The Cupid system [9] implements a generic schema matching algorithm combining linguistic and structural schema matching techniques, and computes normalized similarity coefficients with the assistance of a precompiled thesaurus. The algorithm contains two phases. The first phase, called linguistic matching and the second one is the structural matching of schema elements based on the similarity of their contexts or vicinities. Finally the weighted similarity, a mean of the first and second phases results are calculated. Anchor-PROMPT [2] is another structure base algorithm. It takes as input a set of pairs of related terms—anchors—from the source ontologies. Either the user identifies the anchors manually or the system generates them automatically with the help of string-based techniques, or another matcher computing linguistic (dis)similarity between frame names (labels at nodes) [6]. Then it refines them based on the ontology structures and users' feedback. Anchor-PROMPT traverses the paths between the anchors in the corresponding ontologies. As it traverses the two paths, Anchor-PROMPT increases the similarity score for the pairs of terms in the same positions in the paths. It aggregates the similarity score from all the traversals to generate the final similarity score.

The compositional systems like [12],[4] consist of a set of elementary matchers based on rules, exploiting codified knowledge in ontologies, such as information about super- and sub-concepts, super- and sub-properties, etc. The approach described in [11] is relatively similar to our method. It uses vector characteristics and presents a semantic similarity measure based on a matrix representation of nodes from an RDF labeled directed graph. In this algorithm an entity is described with respect to how it relates to other entities using N-dimensional vectors, N being the number of selected external predicates. Similarities are computed using graph matching algorithm [13]. There are some other methods that benefit from structure of ontologies as well as other techniques such as ola[14], foam[8] and omap[18]. Vector Based Ontology Matching, which we present here, is another vector based model that providing another suggestions for possible matching terms.

## 3 Vector Based Ontology Matching (VBOM)

As mentioned before, the proposed method of ontology matching is based on vector similarity algorithms. Thus, the first step is to model source ontologies in vector notation and then apply a vector matching algorithm to estimate the degree of similarity among them. Similarity of the two vectors can be computed with cosine of angle between those vectors. If the cosine of the angle is 1, the two vectors are exactly the same. As the cosine approaches 0, the similarity degree reduces. Considering $\vec{A}$ and $\vec{B}$ as two vectors, the cosine of their angle can be computed using the following formula:

$$Cos\theta = \frac{\vec{A}.\vec{B}}{\|\vec{A}\|.\|\vec{B}\|} \quad \textbf{(1)}$$

$\vec{A}.\vec{B}$ represents the dot product of two vectors (sum of the product of their corresponding elements). $\|\vec{A}\|$ and $\|\vec{B}\|$ represent the size of the vectors $\vec{A}$ and $\vec{B}$ respectively.

## 3.1 Ontology Vectorization

Ontology Vectorization is the method of modeling two source ontologies (for which the matching problem is of interest) in a single multi dimensional vector space. Any ontology consists of a set of concepts and any concept may have a set of properties which describes that concept. Two types of properties are distinguished:

- ➢ *datatype properties*, relations between instances of classes and RDF literals and XML Schema datatypes.
- ➢ *object properties*, relations between instances of two classes.

The overall perspective of the method is to make a vector space that any of its dimensions represents a unique concept, property or the range of datatype property of the two source ontologies. The vector space must have certain characteristics to be appropriate for utilization in matching algorithm:

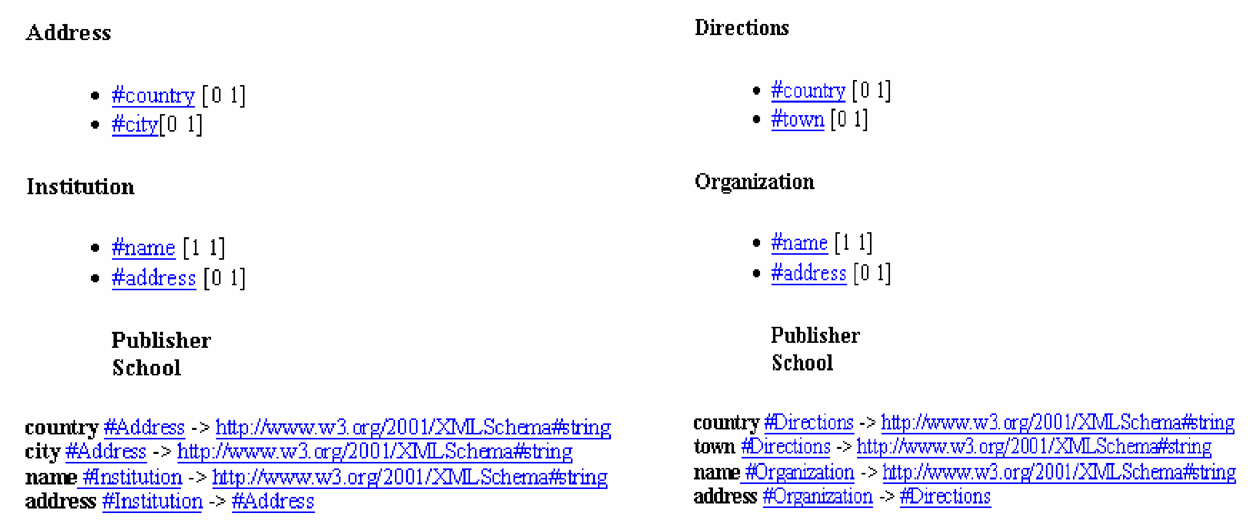
- ✓ Similar concepts, properties and the ranges of datatype properties of the source ontologies will not be duplicated in the vector space.
- ✓ The order of elements is not important. Thus the concepts, properties and the ranges of datatype properties can be arranged in any order for constructing the vector space.
- ✓ The vector space must fully cover all the distinct concepts, properties and the ranges of datatype properties which exist in the two ontologies.

As mentioned before, given a pair of ontologies, vector space is built by extracting all distinct concepts, properties and the ranges of datatype properties belonging to these two source ontologies as its dimensions. Then each of these elements is presented as a vector in this vector space.

Let us have a look at a simple example. Take the following ontologies $O_A$ and $O_B$ in figure 1(the left hand ontology is $O_A$ and the right hand one is $O_B$). The distinct concepts of the two ontologies are: “Address”, “Institution”, “Publisher”, “School”, “Directions”, “Organization”, having “Publisher” and “School” as the subclasses (successors) of “Institution” in $O_A$ and “Organization” in $O_B$. In other words, “Institution” and “Organization” are the ancestors of “School” and “Publisher” in $O_A$ and $O_B$, respectively. The distinct properties are “country”, “city”, “name”, “address” and “town”. Each ontology contains 3 datatype properties and one object property (the values in the brackets show min and max cardinality of the property for that concept). Properties are defined in the following style:

*property Name #domain Name->#range Name.*

Dimensions of our vector space are:{“Address”, “Institution”, “Publisher”, “School”, “Directions”, “Organization”, “country”, “city”, “name”, “address”, “town”, “http://www.w3.org/2001/XMLSchema#string”}. As we mentioned earlier, there is no particular order among the dimensions in the vector space. (Hereafter for simplicity we use “string” instead of http://www.w3.org/2001/XMLSchema#string.)



**Address**

- #country [0 1]
- #city[0 1]

**Institution**

- #name [1 1]
- #address [0 1]

**Publisher**
**School**

**country** #Address -> http://www.w3.org/2001/XMLSchema#string
**city** #Address -> http://www.w3.org/2001/XMLSchema#string
**name** #Institution -> http://www.w3.org/2001/XMLSchema#string
**address** #Institution -> #Address

**Directions**

- #country [0 1]
- #town [0 1]

**Organization**

- #name [1 1]
- #address [0 1]

**Publisher**
**School**

**country** #Directions -> http://www.w3.org/2001/XMLSchema#string
**town** #Directions -> http://www.w3.org/2001/XMLSchema#string
**name** #Organization -> http://www.w3.org/2001/XMLSchema#string
**address** #Organization -> #Directions

**Fig.1.$O_A$ and $O_B$**

Each concept is then described by a vector of weights for itself, all of its properties and ancestors and successors. Furthermore each property is described by a vector of nonzero weights for itself and all of its domain and range concepts.

## 3.2 Weighting Mechanism

**3.2.1 Concept Vectors.** The following shows the weight of each element in the concept vector.

$$W_C(X) = \begin{cases} \log\left(\frac{1}{d_X(c)+1}\right) & if\ d_X(c) > 0 \\ 1 & if\ d_X(c) = 0 \end{cases} \quad \textbf{(2)}$$

Where $W_C(X)$ is the weight of concept *c* in the concept vector *X*, and $d_X(c)$ is the level of distance of concept *c* from *X* in its sub/super class chain. In fact the concept itself acts as a pivot and all of its super/sub classes would receive weights based on their distance from this pivot.

$$W_p(X) = \begin{cases} 1 & if\ X\ \ has\ \ p \\ 0 & if\ X\ \ doesn't\ \ have\ \ p \end{cases} \quad \textbf{(3)}$$

where $W_p(X)$ is the weight of property *p* in the concept vector *X* .

**3.2.2 Property Vectors.**

$$W_C(x) = \begin{cases} 1 & if\ C \in \{xDomain, xRange\} \\ 0 & otherwise \end{cases} \quad \textbf{(4)}$$

where $W_C(x)$ is the weight of concept $c$ in the property vector $x$, *xDomain* is a set of concepts which are the domain of property x, and *xRange* is a concept which is the range of property $x$.

$$W_p(x) = \begin{cases} 1 & if\ p = x \\ 0 & otherwise \end{cases} \tag{5}$$

where $W_p(x)$ is the weight of property $p$ in the property vector $x$.

Consider we want to produce the "Institution" concept vector of $O_A$ in figure 1. As we know "Institution" concept has 2 sub classes: "Publisher" and "School" and 2 properties: "name" and "address". Therefore its vector contains 5 none zero elements: "Institution" "Publisher", "School", "name" and "address". The weight of "Institution" will be 1, the weight of its 2 direct sub classes is $\log(\frac{1}{1+1})$ and the weight of its properties is 1. Thus, according to the vector space which is constructed above, the "Institution" concept vector of $O_A$ is { 0, 1, log (1/2), log (1/2), 0, 0, 0, 0, 1, 1, 0, 0}. Some other concept vectors are: the "Address" concept vector of $O_A$: {1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0}, the "Publisher" concept vector of $O_B$: {0, 0, 1, 0, 0, log (1/2), 0, 0, 0, 0, 0, 0} and so on. Property vectors are also produced. For example "country" property vector of $O_A$ contains 3 none zero elements: its domain ("Address"), itself and its range ("string"). Thus "country" property vector of $O_A$ equals {1,0,0,0,0,0,10,0,0,0,1}. Other vectors are constructed in the same way.

### 3.2 Matching Process

After vectorizing two source ontologies, finding similarities between two ontologies would be easy. As we mentioned in section 3 the correlation between two concept vectors in an N dimensional vector space can be calculated using the cosine of angle between them.

$$sim(\vec{X},\vec{Y}) = \frac{\vec{X}.\vec{Y}}{\|\vec{X}\|.\|\vec{Y}\|} \tag{6}$$

We compute the cosine of all the pairs of concept vectors between the two source ontologies. Then for each concept, we choose the most similar concept with the highest similarity score. This operation is repeated for all the pairs of property vectors.

VBOM is an iterative approach. In each iteration, it selects pairs of similar concepts and similar properties that each participates only in one similarity relation. Then it updates all the vectors of all concepts and properties by setting the weights of participating elements of each selected pair to their biggest non-zero weight. In this way, in each iteration, VBOM benefits from similarities that were discovered in previous iteration. These iterations continue until there are no new similar pairs.

## 4 Results

We carried out experiments on OAEI (Ontology Alignment Evaluation Initiative) 2005 test suite [10]. The evaluation organizers provide a systematic benchmark test suite with pairs of ontologies to align as well as expected (human-based) results. The ontologies are described in OWL-DL and serialized in the RDF/XML format. The expected alignments are provided in a standard format expressed in RDF/XML.

There are different groups of tests in this benchmark [10]:

*Simple tests (tests 1xx).* such as comparing the reference ontology with itself, with another irrelevant ontology or the same ontology in its generalization or restriction to OWL-Lite .

*Systematic tests (tests 2xx).* that are obtained by discarding some features of the reference ontology. (The considered features are names, comments, hierarchy, instances, relations, restrictions, etc.)

- Tests 201 to 210: focus on labels and comments of entities. Names of entities can be replaced by random strings, synonyms, names with different conventions, strings in a language other than English.

- Tests 221 to 247: for these tests the structure is changed. In fact hierarchy can be suppressed, expanded or flattened; properties can be suppressed or their imposed restrictions on classes are discarded and classes can be expanded or flattened.
- Tests 248 to 266: for these tests, names of entities are replaced by random strings; hierarchy can be suppressed, expanded or flattened and properties can be suppressed.

*Four real-life ontologies of bibliographic references (3xx).* that were found on the web and left mostly untouched. These real world ontologies are a combination of complications of the previously mentioned tests.

Table 1. Ontologies with similar labels

| test | Name | Precision | Recall |
|---|---|---|---|
| 101 | Reference | 1 | 1 |
| 102 | Irrelevant Ontology | - | - |
| 103 | Language Generalization | 1 | 1 |
| 104 | Language restriction | 1 | 1 |
| 221 | No specialization | 1 | 1 |
| 222 | Flattened hierarchy | 0.9 | 0.9 |
| 223 | Expanded hierarchy | 1 | 1 |
| 224 | No instance | 1 | 1 |
| 225 | No restrictions | 1 | 1 |
| 228 | No properties | 1 | 1 |
| 230 | Flattened classes | 1 | 1 |
| 231 | Expanded classes | 1 | 1 |
| 232 | | 1 | 1 |
| 233 | | 1 | 1 |
| 236 | | 1 | 1 |
| 237 | | 0.9 | 0.9 |
| 238 | | 0.91 | 0.91 |
| 239 | | 1 | 1 |
| 240 | | 1 | 1 |
| 241 | | 1 | 1 |
| 246 | | 1 | 1 |
| 247 | | 1 | 1 |

We obtained 3 kinds of results in our experiments:

1) Excellent results from ontologies that have similar names (labels) (in tests 1xx, 221 to 247). Because similar names make vectors more similar to each other. In fact the labels are the most important feature to recognize alignments in this approach and if the labels denote an alignment, every thing else can be abandoned. As table 1 shows, both precisions (the number of correct alignments found, divided by the total number of alignments found) and recalls(the number of correct alignments found, divided by the total number of expected alignments) are equal to "1" except for 3 cases;
2) Good results in ontologies are those with similar structures but different naming conventions (in tests 201 to 210 and 249). However the labels are the most important feature in distinction of alignments, the structures of ontologies also play a key role in our approach. We obtained precisions and recalls in the range of 0.78 to 1 and 0.85 to 1 respectively (table 2);
3) Weak results in cases that the two source ontologies are different in both their naming conventions and structures (in tests 248, 250 to 266.). Especially the recall factor is affected more in these situations.(table 3)

Table 2. Ontologies with similar structures and different labels

| test | Name | Precision | Recall |
|---|---|---|---|
| 201 | No names | 0.89 | 0.94 |
| 202 | No names, No comments | 0.89 | 0.94 |
| 203 | No comments | 1 | 1 |
| 204 | Naming conventions | 0.94 | 0.97 |
| 205 | Synonyms | 0.89 | 0.94 |
| 206 | Translation | 0.78 | 0.85 |
| 207 | | 0.89 | 0.94 |
| 208 | | 0.94 | 0.97 |
| 209 | | 0.89 | 0.94 |
| 210 | | 0.89 | 0.94 |
| 249 | | 0.89 | 0.94 |

Table 3. Ontologies with difference in both their labels and structures

| test | Name | Precision | Recall |
|---|---|---|---|
| 248 | | 1 | 0.76 |
| 250 | | 0.6 | 0.09 |
| 251 | | 0.42 | 0.17 |
| 252 | | 0.59 | 0.7 |
| 253 | | 1 | 0.76 |
| 254 | | 0 | 0 |
| 257 | | 0.6 | 0.09 |
| 258 | | 0.42 | 0.17 |
| 259 | | 0.59 | 0.7 |
| 260 | | 0.6 | 0.1 |
| 261 | | 0.4 | 0.06 |
| 262 | | 0 | 0 |
| 265 | | 0.6 | 0.1 |
| 266 | | 0.4 | 0.06 |
| 301 | Real: BibTeX/MIT | 0.73 | 0.53 |
| 302 | Real: BibTeX/UMBC | 0.57 | 0.62 |
| 303 | Real: Karlsruhe | 0.5 | 0.53 |
| 304 | Real: INRIA | 0.84 | 0.9 |

Table 4, depicts summarized results of the three groups of tests and comparison of our method with some other systems. The last row of the Table 1 shows the harmonic mean (H-mean) of three upper values.

Table4. A comparison of VBOM with other systems on OAEI 2005 test suit

| **algo** | **VBOM** | | **foam** | | **omap** | | **ola** | |
|---|---|---|---|---|---|---|---|---|
| test | Prec. | Rec. | Prec. | Rec. | Prec. | Rec | Prec. | Rec |
| 1xx | 1.00 | 1.00 | 0.98 | 0.65 | 0.96 | 1.00 | 1.00 | 1.00 |
| 2xx | 0.81 | 0.74 | 0.89 | 0.69 | 0.31 | 0.68 | 0.80 | 0.73 |
| 3xx | 0.66 | 0.65 | 0.92 | 0.69 | 0.93 | 0.65 | 0.50 | 0.48 |
| H-means | 0.80 | 0.77 | 0.93 | 0.68 | 0.56 | 0.75 | 0.71 | 0.67 |

Although VBOM only focuses on sub/super class chains and properties in ontologies, our experiments show that it is comparable with hybrid models like foam [8] and ola [14] and omap [18] that use linguistic and structural methods. Even in some cases VBOM worked better than the hybrid methods.

VBOM results show that in ontologies that include the sub/super predicate, it is possible to achieve reasonable results by focusing on this predicate and properties. This method is simple and efficient.

## 5 Conclusions

We have presented a structure-based semantic similarity measurement approach for mapping ontologies that can be directly applied to OWL ontologies. The work is based on the intuition that the similarity of two entities can be defined in terms of how these entities are similar with respect to their ancestors, successors and properties. We converted the source ontologies into a vector space of *N* dimensions. These dimensions represent distinct concepts, properties and ranges of datatype properties of two source ontologies. We mapped the concepts in the source ontologies into vectors containing nonzero weights in order to represent their properties and relationships with their ancestors and successors. Also properties are mapped into vectors containing nonzero weights in order to represent their domains and ranges. The results obtained from the tests performed over the Ontology Alignment Evaluation Initiative 2005 test suite are promising. Labels are very important in our approach. After that structures can help the alignment process. In future, we are going to use a dictionary to benefit more from the same labels.

## References

1. Resource description framework. See http://www.w3.org/RDF/.
2. Noy, N. and Musen, M.: Anchor-PROMPT: Using Non-Local Context for Semantic Matching. In Conference on Artificial Intelligence (IJCAI), (2001)
3. Euzenat, J., Valtchev, P.: An integrative proximity measure for ontology alignment, Proceedings of Semantic Integration workshop at ISWC, (2003).
4. Ehrig, M. and Staab, S.: QOM-quick ontology mapping. in proc. 3$^{rd}$ ISWC, Hiroshima (JP), November (2004)
5. Shvaiko, P. and Euzenat J.: A survey of schema-based matching approaches. Journal on Data Semantics, IV, (2005)
6. McGuinness, D.L., Fikes, R., Rice J. and Wilder, S.: An Environment for Merging and Testing Large Ontologies. Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference (KR2000)
7. Owl web ontology language overview. w3c recommendation 10 February 2004. See http://www.w3.org /TR/owl-features/.
8. Ehrig, M. and Sure, Y.: FOAM – Framework for Ontology Alignment and Mapping Results of the Ontology Alignment Evaluation Initiative. Results of the Ontology Alignment Evaluation Initiative. In Integrating Ontologies, (2005)
9. Madhavan, J., Bernstein, P.A. , Rahm, E.: Generic Schema Matching using Cupid. VLDB, (2001).
10. Ontology alignment evaluation initiative. 2005. http://oaei.inrialpes.fr/2005/.
11. Tous, R. and Delgado, J.: A Vector Space Model for Semantic Similarity Calculation and OWL Ontology Alignment. DEXA 2006, pp. 307–316, (2006)
12. Ehrig, M. and Sure, Y.: Ontology mapping - an integrated approach. In Proceedings of the European Semantic Web Symposium (ESWS), pages 76–91, (2004)
13. Blondel, V. D. et al : A measure of similarity between graph vertices: Applications to synonym extraction and web searching. SIAM Rev., 46(4):647–666, (2004)
14. Euzenat, J., Loup, D., Touzani, M. and Valtchev, P.: Ontology Alignment with OLA. In 3rd EON Workshop on Evaluation of Ontology based Tools (EON), Hiroshima, Japan, (2004)
15. Berners-Lee, T.: The semantic web. Scientific American, 284(5):35–43, (2001)
16. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Learning to Map Between Ontologies on the Semantic Web. Proceedings of the 11th international conference on World Wide Web, May 07-11, Honolulu, Hawaii, USA (2002)
17. http://www.worldwidewebsize.com
18. Straccia, U., Troncy, R.: OMAP*:* Combining Classifiers for Aligning Automatically OWL Ontologies. In 6th International Conference on Web Information Systems Engineering (WISE'05), New York City, New York, USA, 133–147, (2005)