A Vector Based Method of Ontology Matching

Zahra Eidoon
ECE Department, University of Tehran, Tehran, Iran
z.eidoon@ece.ut.ac.ir

Nasser Yazdani
ECE Department, University of Tehran, Tehran, Iran
yazdani@ece.ut.ac.ir

Farhad Oroumchian
University of Wollongong , Dubai; POBox 20183, Dubai, UAE
FarhadOroumchian@uowdubai.ac.ae

Abstract

Semantic interoperability is highly influenced by similarities and differences which exist between ontologies. Ontology matching as a solution for finding corresponding concepts among ontologies has emerged to facilitate semantic based negotiations of applications. This paper presents a method of ontology matching which is based on vectorizing ontologies and estimating their similarity degree. A post processing with two heuristic rules also has been employed to improve the results. The proposed method is successfully applied to the test suit of Ontology Alignment Evaluation Initiative 2005 [10] and compared to results obtained by other methods. In general the preliminary results are encouraging and we will continue with the results of some other ontology matchers.

1. Introduction

Ontology is a solution of interoperability problem between heterogeneous data which exist on web. Nowadays in various knowledge domains, there are several overlapping ontologies which differ at the level of abstraction and method of presentation of same concepts [11]. Thus for establishing an efficient communication baseline there is a need for integrating heterogeneous resources of web, and ontology matching is a solution to this problem.

A formal definition of ontology alignment (matching) is: "Given two ontologies O and O', an alignment between O and O' is a set of correspondences (i.e., 4-uples): $\langle e, e', r, n \rangle$ with $e \in O$ and $e' \in O'$ being the two matched entities, r being a relationship holding between e and e',and n expressing the level of confidence [0..1] in this correspondence." [2]

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 later approaches the structural arrangement of elements and their relation to each other is more of interest. Furthermore, ontology matchers can be categorized in 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.

Any ontology consists of a set of concepts that these concepts define a space such that each distinct concept 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.

This paper presents an automatic structural-level ontology alignment technique that is based on a vector matching method. To achieve this goal, an iterative approach has been employed in which vectors representing ontology concepts 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 predicate will be utilized and concepts will be described regarding their ancestors and successors.

The rest of the paper is organized as follows. In section 2 we present Vector Based Ontology Matching (VBOM) method. Results of our method are reported in Section 3. Finally, Section 4 contains some conclusions and future work.

2. 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 ontologies

0-7695-3007-9/07 $25.00 © 2007 IEEE
DOI 10.1109/SKG.2007.75

in vector notation and then apply a vector matching algorithm for estimating the degree of the similarity among them. Optimization of the results will be performed continuously in this technique.

Similarity of the two vectors can be computed with cosine of angle between those vectors. Considering A and B as two vectors, the cosine of their angle can be computed with this formula:

$$\text{Cos } \theta = \frac{A \,.\, B}{\| A \| \,.\, \| B \|}$$

A.B represents dot product of two vectors. ||A|| and ||B|| represent the size of the vector A and B respectively.

2.1. Ontology Vectorization

Ontology Vectorization is the method of modeling two ontologies (for which the matching problem is of interest) in a single multi dimensional vector space. The overall perspective of the method is to make a vector that any of its elements represents a unique concept of ontologies. The vector space must have certain characteristics to be appropriate for utilization in matching algorithm:

- Similar concepts of ontology graphs will not be duplicated in vector space.
- The order of elements is not important. Thus the RDF graphs can be traversed at any order for constructing the vector space.
- The vector space must fully cover all of distinct concepts which exist in two ontologies.

Any ontology consists of a set of concepts .Given a pair of directed labeled RDF graphs of two ontologies, vector space is built by extracting all distinct concepts of these two graphs as its dimensions (i.e. each concept will be a dimension in our vector space and duplicates are discarded). Then each concept is presented as a vector in this vector space. Notice that we assumed the RDF graphs of two ontologies hold the sub/super class relation.

Example1. Let us see a simple example. Take the following graphs in figure 1. G_A representing ontology O_A and G_B representing ontology O_B. The distinct concepts of two ontologies which make our space are: Entity, Book, Chapter, Various and Misk. Although as we mentioned above, the RDF graphs can be traversed at any order for constructing the vector space. For instance {Entity, Book, Chapter, Various, Misk} can be our vector space in this example.

Each concept is then described by a vector contains nonzero weights for itself and all of its ancestors and successors in this way:

1/ (level of distances from concept of interest to other nonzero elements+1)

In fact the concept which we want to make its vector is a pivot and other super/sub classes of it get weights according to their distance from this pivot.

Consider we want produce the Book vector of GA in figure 1. The book vector contains 3 none zero elements: Book, Entity and Chapter. The weight of Book will be 1, the weights of its super class (Entity) is 1/(1+1) and the weight of its subclass is 1/(1+1) too. Thus, the Book vector of GA is :{1/2,1,1/2,0,0}. Some other vectors are: The Book vector of GB : {1/2,1,0,0,0}, The Entity of GA: {1,1/2,1/3,1/2,0,0} and Misk: {1/2, 0, 0, 0, 1}.

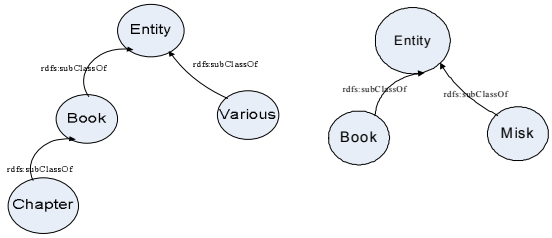

Fig.1. G_A and G_B

Note, if for example concept B has more than one subClassof link (direct or indirect) to concept A, the shortest one is considered.

2.2 Matching Process

After vectorizing two ontologies, matching of concepts should be done. As we mentioned in section 1 the correlation between two vectors in an N dimensional vector space can be calculated using the cosine of angle between them. We compute the cosine of all the pairs of the concept vectors, one from the first source ontology and the other one from the second source ontology, as similarity score. Then for each concept, we choose the most similar concept with highest similarity score. As mentioned before, VBOM is an iterative approach. In each iteration, it finds pairs of similar concepts. Then updates all of the vectors of all the concepts accordingly. For instance if VBOM extracts concept named x of ontology A similar to a concept named y of the ontology B in current iteration, then for all concept vectors of two ontologies, because x and y are similar, their weights should be similar too. Thus for all concept vectors of ontology A, weight of x will be copied to y position and vise versa, for all concept vectors of ontology B , weight of y will be copied to x position.

In this way, in each iteration, VBOM benefits from concepts that were similar in previous iteration. These

iterations are continued until there are no any new extracted similar concept pairs.

2.3 Matching Optimization

VBOM focuses on the chain of super/sub classes for computing similarities. In fact it considers the structures. We apply more structure consideration to optimize our approach and increase the precision factor.

The first optimization rule is in dot product of vectors. In production process, if two weights which should be multiplied together are the same, it means for these two vectors (or concepts) there is a concept which has the same distance to both of them. Therefore in these situations we substitute one for the dot product. For example in figure 1 for computing similarity of the concept Book of the ontology A and the concept Book of the ontology B without optimization we would have:

$$\frac{\{\frac{1}{2},1,\frac{1}{2},0,0\}.\{\frac{1}{2},1,0,0,0\}}{\sqrt{\frac{1}{4}+1+\frac{1}{4}}\times\sqrt{\frac{1}{4}+1}} \approx \frac{\frac{1}{4}+1}{1.37} \approx 0.913$$

Now with the above optimization rule we will have:

$$\frac{\{\frac{1}{2},1,\frac{1}{2},0,0\}.\{\frac{1}{2},1,0,0,0\}}{\sqrt{\frac{1}{4}+1+\frac{1}{4}}\times\sqrt{\frac{1}{4}+1}} \approx \frac{1+1}{1.37} \approx 1.46$$

Of course there is no guarantee that the two concepts with the same weight have similar role in their corresponding ontologies. For example one could be super class and the other sub class. But generally our experiments showed this rule has lead to better results in recall (the number of correct alignments found divided by the total of expected alignments) measure. (See table 1)

The second optimization rule is in cases where one concept in ontology A has more than one similar concept in the ontology B with the same exact similarity score. In these conditions VBOM compares number of attributes and super/sub classes of concepts.

Definition: if a concept (named x) from ontology A is similar to more than one concept from ontology B, Select concept y from ontology B as similar if y has the same number of attributes and super/sub classes as x. If still there is more than one similar concept to x, then select the concept y based on the number of siblings.

This rule improves our precision (the number of correct alignments found divided by the total number of alignments found) better. (See table 1)

3. Results

To test our approach we have used the 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 and described in [10].

In Table 1 and 2, rows correspond to the test numbers, while the columns correspond to the obtained values of precision and recall. Table 1 shows the precision and recall values for the vector only approach (VBOM) and VBOM with only heuristic rule1 or 2 and VBOM with both heuristic rules.

Table 1. The good impression of VBOM optimization rules on OAEI 2005 test cases

test	VBOM without opt.rules Prec. rec.	VBOM with opt. rule 1 prec. rec.	VBOM with opt. rule 2 prec .rec.	VBOM with both rules prec. rec
205	0.47 0.21	0.21 0.72	1.0 0.21	0.71 0.73
209	0.47 0.21	0.21 0.72	1.0 0.21	0.71 0.73
230	1.0 1.0	1.0 1.0	1.0 1.0	1.0 1.0
260	0.5 0.07	0.67 0.13	0.7 0.07	0.66 0.14
265	0.5 0.07	0.67 0.13	0.7 0.07	0.66 0.14
266	0.29 0.06	0.67 0.12	0.67 0.06	1.0 0.06

Based on these results, the heuristic rule 1 improves the recall but the heuristic rule 2 increases the precision. The combination of two creates a balance between recall and precision.

Table 2, depicts a comparison of the VBOM method with both optimization rules. Although VBOM only focused on sub/super class chains in ontologies, our experiments show that it is comparable with hybrid models like VSM[11], FOAM[8] and OLA[14] that use linguistic and structural methods. Even in some cases VBOM worked better than the hybrid methods.

VBOM results shows that in ontologies that include the sub/super predicate, it is possible to achieve reasonable results by only focusing on this predicate in RDF labeled directed graph. This method is simple and efficient.

We obtained better results in ontologies that contain similar concept names and similar structures (e.g. tests 205,209,230,103,104,203 and 204). Because similar concept names make vectors more similar to each other

and similar structures help to obtain the bigger results from dot product according to our optimization rule1. In cases the two ontologies are very different in their naming convention or structures, the results were less precise (e.g. tests 201,260,265 and 266).Specially the recall was affected more in these situations. VBOM is applicable only on ontologies that have a hierarchical structure and this is the limitation of this approach. But since most of the ontologies are organized in hierarchy structures, it is not a major limitation.

Table 2. Comparisons of VBOM with both heuristic rules and three other methods on the OAEI 2005 test collection.

test	VBOM prec.	VBOM rec.	VSM prec.	VSM rec.	FOAM prec.	FOAM rec.	OLA Prec.	OLA rec.
205	0.71	0.73	0.90	0.89	0.89	0.73	0.43	0.42
209	0.71	0.73	0.88	0.87	0.78	0.58	0.43	0.42
230	1.0	1.0	0.97	0.96	0.94	1.0	0.95	0.97
260	0.66	0.14	0.44	0.42	0.75	0.31	0.26	0.17
265	0.66	0.14	0.44	0.42	0.75	0.31	0.22	0.14
266	1.0	0.06	0.45	0.42	0.67	0.36	0.14	0.09
103	1.0	1.0						
104	1.0	1.0						
201	0.67	0.12						
203	1.0	1.0						
204	0.78	0.85						
206	0.54	0.73						

4. Conclusion

We have presented here an approach to structure-based semantic similarity measurement that can be directly applied to OWL ontologies modeled as RDF labeled directed graphs. The work is based on the intuition that similarity of two entities can be defined in terms of how these entities are similar with respect to their ancestors and successors. We modeled these relationships with a vector space of N dimensions, N being the number of distinct concepts of two ontologies. We map the concepts in the ontologies into vectors contain nonzero weights to represent their relationships with their ancestors and successors. We have also presented two heuristic rules for optimization of matching results. The results obtained in the tests performed over the Ontology Alignment Evaluation Initiative 2005 test suite are encouraging.

In future we are going to experiment with different approaches of vector matching. We will also try to use other predicates than super/sub class predicates.

References

[1]. Resource description framework. See http://www.w3.org/RDF/.

[2]. M. Ehrig and J. Euzenat, "Relaxed precision and recall for ontology matching", In Proc. K-Cap workshop on Ontology integration, Banff (CA), 2005, pp. 25–32.

[3]. N. Noy and M. Musen, "Anchor-PROMPT: Using Non-Local Context for Semantic Matching", In Conference on Artificial Intelligence (IJCAI), 2001.

[4]. M. Ehrig and S. Staab. "Qom - quick ontology mapping", In Proceedings of the International Semantic Web Conference (ISWC), 2004, pp. 683–697.

[5]. P. Shvaiko and J. Euzenat, "A survey of schema-based matching approaches",*Journal on Data Semantics*, IV, 2005.

[6]. D.L. McGuinness, R. Fikes, J. Rice and S. Wilder, "An Environment for Merging and Testing Large Ontologies, Proceedings of the Seventh International Conference (KR2000). 2000.

[7]. Owl web ontology language overview. w3c recommendation 10 February 2004.

[8]. M. Ehrig and Y. Sure, "FOAM – Framework for Ontology Alignment and Mapping Results of the Ontology Alignment Evaluation Initiative", Proceedings of the Workshop on Integrating Ontologies,2005, pp. 72–76

[9]. J. Madhavan, P.A. Bernstein, E. Rahm, "Generic Schema Matching using Cupid", VLDB 2001.

[10]. Ontology alignment evaluation initiative. 2005.

[11]. R. Tous and J. Delgado, "A Vector Space Model for Semantic Similarity Calculation and OWL Ontology Alignment", DEXA 2006,2006, pp. 307–316.

[12]. M. Ehrig and Y. Sure, "Ontology mapping - an integrated approach, In Proceedings of the European Semantic Web Symposium (ESWS), 2004, pp. 76–91.

[13]. V. D. Blondel et al, "A measure of similarity between graph vertices: Applications to synonym extraction and web searching", SIAM Rev., 2004.

[14]. J. Euzenat, D. Loup, M. Touzani, and P. Valtchev, "Ontology Alignment with OLA", In 3rd EON Workshop, 3rd Int. Semantic Web Conference, 2004, pp. 333–337.