Result of Ontology Alignment with RiMOM at OAEI'07

Yi Li, Qian Zhong, Juanzi Li, and Jie Tang
Department of Computer Science and Technology, Tsinghua University
{ly, zhongqian, ljz, tangjie}@keg.cs.tsinghua.edu.cn

Abstract. In this report, we give a brief explanation of how RiMOM obtains the ontology alignment results at OAEI'07 contest. RiMOM integrates different alignment strategies: edit-distance based strategy, vector-similarity based strategy, path-similarity based strategy, background-knowledge based strategy, and three similarity-propagation based strategies. Each strategy is defined based on one specific ontological-information. In this contest, we, in particular, study how the different strategies (or strategy combination) perform for different alignment tasks. We found that: 1) on the directory data set, the path-similarity based strategy seems to outperform the others and 2) on the anatomy and food data sets, the background-knowledge based strategy has several distinct advantages. This report presents our results based on the evaluation. We also share our thoughts on the experiment design, showing specific strengths and weaknesses of our approach.

1. PRESENTATION OF THE SYSTEM

Ontology alignment is the key point to reach interoperability over ontologies. In recent years, much research work has been conducted for finding the alignment of ontologies [1] [4].

We have studied different strategies for ontology alignment and implemented them in a tool called RiMOM [5]. Each strategy is defined based on one kind of ontological information. In total, there are more than seven strategies implemented in RiMOM, we investigate the difference between the strategies and study which strategy will obtain the best performance on a specific alignment task. This introduces a very interesting (also critical) research issue: how to find a best strategy or an optimal strategy combination given an alignment task, called strategy selection.

1.1 State, purpose, general statement

For simplifying the following description, we here define the notations used throughout the report.

Ontology: An ontology O is composed of concepts C, properties/relations R, instances I, and Axioms A^O. We here use capital letter to indicate a set and lowercase letter (e.g., $c \in C$) to indicate one element in the set. Sometimes, for further simplification, we use entity e to indicate either c or r.

Ontology alignment: given an alignment from ontology O_1 to O_2, we call ontology O_1 as source ontology and O_2 as target ontology. We call the process of finding the alignment from O_1 to O_2 as (Ontology) alignment discovery or alignment finding.

Challenges for automating ontology alignment include: 1) how to automatically find alignments of high quality; 2) how to find the alignments efficiently; 3) what is the difference between the various alignment strategies and which one should be used for a specific task; 4) how to deal with the alignment of large scale ontology; 5) how to ease parameterizing, as the accuracy of alignments may vary largely with different parameters; 6) how to make full use of the user interaction.

In this campaign, we focus on dealing with the problems of 1), 2), and 3) with our system RiMOM.

1.2 Specific techniques used

There are six major steps in a general alignment process of RiMOM:

1) Similarity factors estimation. Given two ontologies, it estimates two similarity factors, which respectively approximately represent the structure similarity and the label similarity of the two ontologies. The two factors are used in the next step of strategy selection.

2) Strategy selection. The basic idea of strategy selection is that if two ontologies have high label similarity factor, then RiMOM will rely more on linguistic based strategies; while if the two ontologies have high structure similarity factor, then we will employ similarity-propagation based strategies on them. See Section 1.2.1 for details. Strategy selection by the two factors is mainly used on the benchmark data set. For the directory, anatomy, and food data set, we chose the strategies manually.

3) Single strategy execution. We employ the selected strategies to find the alignment independently. Each strategy outputs an alignment result.

4) Alignment combination. It combines the alignment results obtained by the selected strategies. The combination is conducted by a linear-interpolation method.

5) Similarity propagation. If the two ontologies have high structure similarity factor, RiMOM employs a similarity propagation process to refine the found alignments and to find new alignments that cannot be found using other strategies.

6) Alignment refinement. It refines the alignment results from the previous steps. We defined several heuristic rules to remove the "unreliable" alignments.

1.2.1 Similarity factors estimation

Our preliminary experiments show that the multi-strategy based alignment does not always outperform its single-strategy counterpart. For a new, unseen mapping task, we propose to use two similarity factors to determine which strategy should be used.

Given two ontologies: source ontology O_1 and target ontology O_2, we calculate two approximate similarity factors: structure similarity factor and label similarity factor.

We define structure similarity factor as: $F\_SS = \frac{\#common\_concept}{\max(\#nonleaf\_c_1, \#nonleaf\_c_2)}$, where $\#nonleaf\_c_1$ indicates the number of concepts in O_1 that has sub concepts. Likewise for $\#nonleaf\_c_2$. *#common_concept* is calculated as follows: if concepts $c_1 \in O_1$ and $c_2 \in O_2$ have the same number of sub concepts and they are in the same depth from

the concept "owl:Thing", we add one to *#common_concept*. After enumerated all pair, we obtain the final score of *#common_concept*. Intuition of the factor is that the larger the structure similarity factor, the more similar the structures of the two ontologies are.

The label similarity factor is defined as: $F\_LS = \frac{\#same\_label}{\max(\#c_1, \#c_2)}$, where $\#c_1$ and $\#c_2$ respectively represent the number of concepts in O_1 and O_2. *#same_label* represents the number of pairs of concepts $\{(c_1, c_2) | c_1 \in O_1$ and $c_2 \in O_2\}$ that have the same label.

The two factors are defined simply and not used to accurately represent the real "similarities" of structures and labels. However, they can approximately indicate the characteristics of the two ontologies. Moreover, they can be calculated efficiently.

So far, we carried out the strategy selection by heuristic rules. For example, if the structure similarity factor *F_SS* is lower than 0.25, then RiMOM suppresses the CCP and PPP strategies. However, the CPP will always be used in the alignment process.

1.2.2 Multiple strategies

The strategies implemented in RiMOM include: edit-distance based strategy, vector-similarity based strategy, path-similarity based strategy, background-knowledge based strategy, and three similarity-propagation based strategies.

1. Edit-distance based strategy (ED)

Each label (such as concept name or property name) is composed of several tokens. In this strategy (ED), we calculate the edit distance between labels of two entities. Edit distance estimates the number of operations needed to convert one string into another. We define (1-*#op*/*max_length*($l(e_1)$, $l(e_2)$)) as the similarity of two labels, where #op indicates the number of operations, *max_length*($l(e_1)$, $l(e_2)$) represents the maximal length of the two labels.

2. Vector-similarity based strategy (VS)

We formalize the problem as that of document similarity. For an entity *e*, we regard its label, comment, and instances as a 'document' and calculate the similarity between an entity pair. Specifically, the 'document' is tokenized into words. Then we remove the stop words and employ stemming on the words and view the remains as features to generate a feature vector. We also add some other general features which prove to be very helpful. For a concept, the features include: the number of its sub concepts, the number of properties it has, and the depth of the concept from "OWL:Thing". Next, we compute the cosine similarity between two feature vectors. The advantage of this strategy is that it can easily incorporate different information (even structural information) into the feature vector.

3. Path-similarity based strategy (PS)

We define path as the aggregation of the entity labels from "OWL:Thing" to the current entity. A path-similarity measure between two entities e_1 and e_2 is defined as:

$$sim_p(e_1, e_2) = \max(sim(l(e_1), PL(e_2)), sim(PL(e_1), l(e_2)))$$

where $PL(e_2)$ is the path of e_2. $sim(l(e_1), PL(e_2))$ is the similarity between the label of entity e_1 and the path of entity e_2. It is estimated by averaging similarities between the label of e_1 and each label in the path of e_2.

4. Background-knowledge based strategy (BK)

We also try to make use of background knowledge to enhance the performance of alignment. The idea is straightforward. In some alignment tasks, for example the food alignment task and the anatomy alignment task, the available information is limited (only concept labels are available). We utilize the available knowledge base (we used wiki pages) to help find the alignment. For each entity, we first look up in the knowledge base for its definition, and then use the description of its definition in the similarity calculation of the vector-similarity based strategy.

5. Strategy combination

For some alignment task, we need use more than one strategy to find the alignment. The strategies are employed first independently and then are combined together. A combination measure is thus defined as:

$$Map(e_1, e_2) = \frac{\sum_{k=1...n} w_k \sigma\left(Map_k(e_1, e_2)\right)}{\sum_{k=1...n} w_k}$$

where $e_1 \in O_1$ and $e_2 \in O_2$; $Map_k(e_1,e_2)$ is the alignment score obtained by strategy k. w_k is the weight of strategy k. σ is a sigmoid function, which is defined as $\sigma(x) = 1/\left(1+e^{-5(x-\alpha)}\right)$, where α is tentatively set as 0.5.

This "independence-and-combination" fashion has the advantage of easy integrating new strategies into the alignment process.

6. Similarity-propagation based strategies

The structure information in ontologies is useful for finding the alignments especially when two ontologies share the common/similar structure. According to the propagation theory [2], we define three structure based strategies in RiMOM, namely concept-to-concept propagation strategy (CCP), property-to-property propagation strategy (PPP), and concept-to-property propagation strategy (CPP).

Intuition of the propagation based method is that if two entities are aligned, their super-concepts have higher probability to be aligned. The basic idea here is to propagate the similarity of two entities to entity pairs that have relations (e.g., subClassOf, superClassOf, siblingClassOf, subPropertyOf, superPropertyOf, range, and domain) with them. The idea is inspired by similarity flooding [3]. We extended the algorithm and adaptively used them in the three structure based strategies.

In CCP, we propagate similarities of concepts pair across the concept hierarchical structure. In PPP, we propagate similarities of property pair across the property hierarchy. In CPP, we propagate similarities of concepts pair to their corresponding property pair, and vice versa. Details of the method will be reported elsewhere.

The similarity-propagation based strategies are performed after the other strategies defined above. They can be used to adjust the alignments and find new alignments.

1.3 Adaptations made for the evaluation

Some parameters were tuned and set in the experiments. For example, for strategies combination (cf. equation 1), we set the weight of ED as 0.5 and that of VS as 1. For strategy selection, we define 0.25 as the threshold to determine whether CCP and PPP will be suppressed or not. We also define 0.2 as threshold to determine whether ED

will be suppressed or not. In addition, we employed background-knowledge based strategy for food and anatomy alignment, and path-similarity based strategy for directory.

1.4 Link to the system, parameters file, and provided alignments

Our system RiMOM (including the parameters file) can be found at http://keg.cs.tsinghua.edu.cn/project/RiMOM/. For details of the approach, see [5].

The alignment results of the campaign are available at http://keg.cs.tsinghua.edu.cn/project/RiMOM/OAEI2007/.

2 Results

RiMOM has been implemented in Java. We use OWL-API to parse the RDF and OWL files. The experiments were carried out on a Server running Windows 2003 with two Dual-Core Intel Xeon processors (2.8 GHz) and 3-gigabyte memory. All the alignments outputted by RiMOM are based on the same parameters.

2.1 Benchmark

There are in total 54 alignment tasks defined on the benchmark data set. The task is to find the alignment from every ontology to the reference ontology 101. We conducted alignment on the benchmark data set in the following steps: 1) we first employ the vector-similarity based strategy. We make use of the entity labels, comments, and instances to generate a feature vector and calculate the similarity between each entity pair; 2) we utilize the similarity-propagation based strategies to refined alignment results.

We also compute the similarity factors and use the similarity factor in Step 1) (for determining whether we add special features into the feature vector) and 2) (for determining whether a propagation based strategy should be used).

In these tasks, the average precision is 0.97 and the average recall is 0.99. The average time cost is about 4 second per task.

2.2 directory

The directory ontologies are organized as a taxonomy with sub-sumption hierarchies. We obtain the alignment results in the following ways: 1) edit-distance based strategy is used to calculate the similarity between entity labels; 2) path-similarity based strategy is employed to compute the similarity between two entity paths; 3) combination of the two similarities; 4) similarity propagation (CCP) is utilized on the hierarchical structure to refine the result; and 5) pruning some found alignments. The pruning is performed using heuristic rules.

2.3 anatomy

For anatomy ontology, we utilize the background-knowledge based strategy to find the alignment. Specifically, we perform the alignment finding in the following steps: 1) we constructed a background knowledge base by using the concept definitions from UMLS [6], in total we have a base of more than 100 K terms; 2) for each entity from the source ontology, we find if there is an entity with the identical label in the target ontology. If so, we alignment them; otherwise, we look up in the background knowledge base to find the definition description of the label; 3) we use the vector-similarity based strategy to calculate the similarity. We create the feature vector using the entity label and the concept definition (if found in the knowledge base).

2.4 food

We employ the same process as that in anatomy to find alignment on the food data set by using wiki as the background-knowledge.

3 General comments

3.1 Comments on the results

An objective and comprehensive comment on strengths or weakness requires the comparison with other participants, which are not available so far (will be available before the workshop). Here, we share some thoughts about the results.

Strengths

From experimental results, we see that RiMOM can achieve high performance when the ontologies to be aligned have similar linguistic information or similar structure information. Some concluding remarks are summarized as follows:

1) Linguistic information (including label of concepts and properties) is important and help to align most of the entities.

2) Structure information can be used to improve the alignments, in particular when linguistic information is missing.

3) Strategy selection is important. In different alignment tasks, the ontologies to be aligned have different characteristics, it would be particularly helpful to find the characteristics of the ontologies and apply correspondingly strategies on them. This also introduces an interesting research issue: how to perform the strategy selection efficiently? Currently, we use two factors to select the structure strategy and to determine whether we add several features into the vector when using vector-similarity based strategy. However, it is far from an ideal solution of the strategy selection.

4) Alignment refinement is helpful. We removed the unreliable alignments.

Weakness

1) Although the preliminary experiments show that our strategy selection method can enhance the alignment finding, it is not sufficient. There are many problems needed to be solved.

2) We note that parameter setting is very important. We have found that using different parameter settings, with the exactly same approach, the alignment results may differ largely. So far, we tuned the parameters manually. It is not adaptable in particular when the ontologies are very large, which means that tuning different parameters to find the best ones is not possible.

3.2 Discussions on the way to improve the proposed system

Possible improvements are corresponded to the related weaknesses in the previous section.

1) New strategy selection by considering all the strategies and all the factors should be proposed.

2) Our thinking is to use a supervised machine learning method to find the optimal parameters based on some training data sets.

3.3 Comments on the OAEI 2007 test cases

The benchmark tests indicate very interesting general results on how the alignment approach behaves. These tests are really useful, as a good underlying test base, for evaluating and improving the alignment algorithm and system.

For future work, it might be interesting to add some tests to evaluate the cross-linguistic alignment, as for English ontology to Chinese ontology, an issue is important in practical application.

4 Conclusion

In this report, we have briefly introduced how we employed RiMOM to obtain the alignment results in OAEI'07 contest. We have presented the alignment process of RiMOM and explained the strategy defined in RiMOM. We have also described how we performed the alignment for different alignment tasks. We summarized the strengths and the weaknesses of our proposed approach and gave possible improvement for the system in the future work.

Acknolwegement

The work is supported by the National Natural Science Foundation of China under Grant No. 90604025 and No. 60703059. It is also supported by IBM Innovation funding.

References

[1] J. Euzenat. State of the art on ontology alignment. http://www.inrialpes.fr/exmo/cooperation/kweb/heterogeneity/deli/. August, 2004.

[2] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. International Journal of Computer Vision, Vol. 70, No. 1, October 2006.

[3] S. Melnik, H. Garcia-Molina and E. Rahm: Similarity Flooding: a versatile graph matching algorithm and its application to schema matching. In Proc. of 18th ICDE. San Jose CA, Feb 2002. pp. 117-128

[4] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. The VLDB Journal, 2001, 10:334-350.

[5] J. Tang, J. Li, B. Liang, X. Huang, Y. Li, and K. Wang. Using Bayesian Decision for Ontology Alignment. Journal of Web Semantics, Vol(4) 4, pp. 243-262, 2006.

[6] http://umlsks.nlm.nih.gov/kss/

Appendix: Raw results

The following results were obtained in the evaluation runs.

Matrix of results

#	Name	Prec.	Rec.
101	Reference alignment	1.00	1.00
102	Irrelevat ontology	N/A	N/A
103	Language generalization	1.00	1.00
104	Language restriction	1.00	1.00
201	No names	1.00	1.00
202	No names, no comments	1.00	0.80
203	No comments	1.00	0.88
204	Naming conventions	1.00	1.00
205	Synonyms	1.00	0.99
206	Translation	1.00	0.99
207		1.00	0.99
208		0.98	0.86
209		1.00	0.84
210		0.99	0.85
221	No specialisation	1.00	1.00
222	Flatenned hierachy	1.00	1.00
223	Expanded hierarchy	1.00	1.00
224	No instance	1.00	0.99
225	No restrictions	1.00	1.00
228	No properties	1.00	1.00
230	Flatenned classes	0.94	1.00
231		1.00	1.00
232		1.00	0.99
233		1.00	1.00
236		1.00	1.00
237		1.00	0.99

238		1.00	0.99
239		1.00	1.00
240		1.00	1.00
241		1.00	1.00
246		1.00	1.00
247		1.00	1.00
248		0.99	0.78
249		1.00	0.79
250		1.00	0.55
251		0.76	0.58
252		0.85	0.70
253		0.99	0.77
254		1.00	0.27
257		1.00	0.55
258		0.76	0.57
259		0.85	0.69
260		0.93	0.45
261		1.00	0.27
262		1.00	0.27
265		0.93	0.45
266		1.00	0.27
301	BibTeX/MIT	0.75	0.67
302	BibTeX/UMBC	0.72	0.65
303	Karlsruhe	0.45	0.86
304	INRIA	0.90	0.97