RiMOM Results for OAEI 2008

Xiao Zhang[1], Qian Zhong[1], Juanzi Li[1], Jie Tang[1], Guotong Xie[2] and Hanyu Li[2]
[1]Department of Computer Science and Technology, Tsinghua University, China
{zhangxiao,zhongqian,ljz,tangjie}@keg.cs.tsinghua.edu.cn
[2]IBM China Research Lab
{XIEGUOT, lihanyu}@cn.ibm.com

Abstract. In this report, we give a brief explanation of how RiMOM obtains the ontology alignment results at OAEI 2008 contest. We introduce the alignment process of RiMOM and more than 8 different alignment strategies integrated in RiMOM. Since every strategy is defined based on one specific ontological-information, we, in particular, study how the different strategies perform for different alignment tasks in the contest and design a strategy selection technique to get better performance. The result shows this technique is very useful. We also discuss some future work about RiMOM.

1 Presentation of the system

Ontology matching is the key technology to reach interoperability over ontologies. In recent years, much research work has been conducted for finding the alignment of ontologies [1] [2].

RiMOM [3] is an automatic ontology matching system implemented in JAVA. In RiMOM, we implement several different matching strategies. Each strategy is defined based on one kind of ontological information. Moreover, we investigate the differences between the strategies and compare the performances of different strategies on different matching tasks. One of the most important issues we introduce in RiMOM is how to choose appropriate strategies (or strategy combination) according to the features and the information of the ontologies.

1.1 State, purpose, general statement

For simplifying the following description, we here define the notations used throughout the report. An ***ontology*** *O* is composed of concepts *C*, properties/relations *R*, instances *I*, and Axioms A^o. We here use capital letters to indicate a set and lowercase letters (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*.

We implement more than 8 different strategies in RiMOM. Experiments show that the multi-strategy based alignments do not always beat its single strategy counterpart. We define three ontology feature factors: Label Similarity Factor (LF), Structure Similarity Factor (SF) and Label Meaning Factor (MF) for strategy selection. The definition of the three factors can be found in 1.2.1.

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

1) Ontology feature factors estimation. Given two ontologies, it estimates three ontology feature factors. The three 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 base strategies on them. Moreover, if the labels are full of semantic, we will use WordNet [4] based strategy instead of Edit-distance based strategies. We also use these factors to decide the thresholds when refining the results. Strategy selection is mainly used on the benchmark data set. For the directory, mldirectory, anatomy, and fao data set, we choose 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 Specific techniques used

1.2.1 Ontology Feature factors estimation

Given two ontologies: source Ontology O_1 and target ontology O_2, we calculate three ontology feature factors, including two approximate similarity factors between two ontologies (Structure Similarity Factor and Label Similarity Factor) and one factor representing the semantics of entity labels in each ontology (Label Meaning Factor) .

We define structure similarity factor as: $SF = \frac{\#common\_concept}{\max(\#nonleaf\_O_1, \#nonleaf\_O_2)}$, where $\#nonleaf\_O_1$ indicates the number of concepts in O_1 that has sub concepts. Likewise for $\#nonleaf\_O_2$. $\#common\_concept$ 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 root concept, we add one to $\#common\_concept$. After enumerating 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: $LS = \frac{\#same\_label}{\max(\#c_1, \#c_2)}$, 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 that have the same label.

The label meaning factor is defined as: $MF = \frac{\#label\_with\_meaning}{\#entity}$, where $\#label\_with\_meaning$ represents the number of entities whose label is meaningful, and $\#entity$ represents the number of entities in the ontology. We use WordNet to judge whether a label is meaningful or not.

Now the three factors are defined very simply. The first two factors are 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 Factor MF is larger than 0.9, then RiMOM uses WordNet based strategy instead of edit-distance based strategy. If the structure similarity factor SF 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-based similarity based strategy, path-similarity based strategy, dynamic path-similarity based strategy, Japanese-English path-similarity strategy and similarity-propagation based strategies.

1. Edit-distance based strategy(ED)

Each label (such as concept names or property names) is composed of several tokes. In this strategy, we calculate the edit distance between labels of two entities. Edit distance estimates the number of operation needed to convert one string into another. We define $(1 - \#op / \text{max\_length}(l(e_1), l(e_2)))$ as the similarity of two labels, where $\#op$ indicates the number of operations, $\text{max\_length}(l(e_1), l(e_2))$ represents the maximal length of the two labels.

2. WordNet based strategy (WN)

In this strategy, RiMOM first preprocesses each label into a bag of words. When calculate the similarity from one bag of words to another, for every word in the first bag, RiMOM find the most similar word in the second with WordNet, then calculate the mean of the similarities as the similarity from the first bag to the second. The similarity of the two labels is the mean of the similarity of two bags of words in two directions.

3. Vector-similarity based strategy(VS)

We formalize the problem as that of document similarity. For any entity e, we regard its label, comment, and instances as a “document” and calculate the similarity between entities. Specially, 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 the root concept. 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.

4. Path-similarity based strategy (PS)

We define the path of labels as the aggregation of the entity labels from the root entity to the current entity. The paths of the labels of the two entities can be represented as $L_1 = a_1 a_2 .. a_m$ and $L_2 = b_1 b_2 .. b_n$. The path-similarity measure between two entities e_1 and e_2 is defined as:

$$Sim(e_1, e_2) = \sum_{i=1}^{m-1} w_i \bullet \max_{j=1}^{n-1}(LabelSim(a_i, b_j)) + w_m \bullet LabelSim(a_m, b_n)$$

The $LabelSim(a_i, b_j)$ is calculated using either edit-distance or WordNet.

5. Dynamic path-similarity based strategy (DPS)

The path of labels can also be considered as a path of entities, especially when the main information of the ontology is the labels. We have three assumption for this strategy: 1) for the two path of entities, we always match from the short path to the long one, and every entity in the short path can be matched to an entity in the long one; 2) no matched pairs are “crossed”, that is to say, the matching result is consistent with the hierarchy represented in the path; 3) when calculating the similarity of current pair of entities, the matching result of the prev-path is optimal. Then we can calculate the similarity of two paths of entities using the dynamic programming technique.

6. Strategy combination

For some alignment task, we need use more than one strategy to find the alignment. The strategies are employed first independently to calculate the similarity between entities and the similarities are combined together. Our combination measure is defined as:

$$Sim(e_1, e_2) = \frac{\sum_{k=1}^{n} w_k \sigma(Sim_k(e_1, e_2))}{\sum_{k=1}^{n} w_k}$$

Where $e_1 \in O_1$ and $e_2 \in O_2$. $Sim_k(e_1, e_2)$ is the alignment score obtained by strategy *k*. w_k is the weight of strategy *k*. For vector similarity based strategy, the weight is always 1 while for WordNet and edit-distance based strategies, the weight is generated automatically. σ is sigmod function, which is defined as $\sigma(x) = 1/\left(1 + e^{-5(x-\alpha)}\right)$, where α is tentatively set as 0.5.

7. 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 [7], we define three structure based strategies in RiMOM, namely concept-to-concept propagation strategy (CCP), property-to-property propagation strategy (PPP), and the concept-to-property propagation strategy (CCP).

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 [8]. 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. Likely, we propagate similarities of property pair across the property hierarchy in PPP and concepts pair to their corresponding property pair in CPP.

Furthermore, there are some object properties in the ontologies which may have the similar characteristics with subClassOf property. Every pair of concepts with such property has a relation similar to sub-super concept relation. However, these pairs of concepts are usually manipulated as the domain and range of property and the relation is lost. RiMOM can also use these properties for similarity-propagation.

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

8. Indirect Matching

We also use the indirect matching technique in RiMOM. It is sometimes very difficult to match two ontologies directly. Since the source ontology and the target ontology are usually concerned with the same domain of knowledge, it is possible to match both the source and target ontology to a third one. Then the entities in the source and target ontology which match to the same entity in the third ontology can be aligned. RiMOM can take three ontologies as input and execute the indirect matching.

1.3 Adaptations made for the evaluation

Some parameters are tuned and set in the experiments. For example, for strategy selection, we define 0.25 as threshold to determine whether CCP and PPP will be suppressed or not. We also define MF factor threshold as 0.9 to determine whether use WordNet based strategy instead of edit-distance based strategy. In addition, we employ dynamic path similarity for directory task and path-similarity based strategy for mldirectory task.

1.4 Link to the system , parameters file and the set of provided alignments.

Our system RiMOM (RiMOM does not need the parameters file) can be found at http://keg.cs.tsinghua.edu.cn/project/RiMOM/.
The alignment results of the campaign are available at
http://keg.cs.tsinghua.edu.cn/project/RiMOM/OAEI2008/.

2 Results

RiMOM has participated in 5 tasks in OAEI 2008, including benchmark, anatomy, fao, directory and mldirectory. RiMOM use OWL-API to parse the RDF and OWL files. The experiments are carried out on a PC running Window XP with AMD Athlon 64 X2 4200+ processor (2.19GHz) and 2G memory.

2.1 benchmark

There are in total 111 alignment tasks defined on the benchmark data set. RiMOM takes exactly the same steps introduced in 1.1. However, on the tasks where the labels are absolutely random strings, the WordNet based strategy and edit-distance based strategy are suppressed. The vector-similarity based strategy is always employed.

RiMOM get perfect alignment in the 101, 103, 104 tests. RiMOM also do quite well in the 2xx tests. Except the data sets in which almost all the information are suppressed like 26x and 25x, RiMOM aligns the source and target ontology with both good precision and recall. Even in those data set most information missing, RiMOM still can find some alignments with very high precision. Compared to the result of OAEI 2007, RiMOM also improve the performance in the real ontology data sets 301, 302, 303, 304.

2.2 anatomy

The anatomy data set contains two large scale anatomy ontologies. RiMOM employs edit-distance based strategy on labels to get the initial mapping, then employs both the concept-to-concept propagation strategy and the propagation strategy on the object property "UNDEFINED_part_of" to get the alignments which cannot be extracted by just comparing the labels simply. The propagation strategy can find about 15% more alignments.

2.3 fao

The scale of the fao data set is even larger than the anatomy data set, so we only use the edit-distance based strategy on labels to calculate the similarity. Moreover, because the FAO ontology is better formed than larger than the other two, we use the FAO ontology as a standard ontology to indirectly match the AGROVOC ontology and ASFA ontology.

2.4 directory

As all the ontologies in directory data set are in the "chain" form, RiMOM just employs the dynamic path-similarity based strategy to get the similarity matrix. Then RiMOM extracts the alignments with no "crossed" matched entity pairs.

2.5 mldirectory

The mldirectory data set is composed of three kinds of tasks: the matching between English ontologies, the matching between Japanese ontologies and English ontologies and the matching between the Japanese ontologies. For this task, RiMOM mainly depends on the ID of the concepts and the hierarchical information. When dealing with the Japnanese IDs, we takes the following preprocessing steps: 1) use the tool named ChaSen [5] for segmentation of Japanese IDs; 2) use the dictionary JMDict [6] to translate the Japanese words into English; 3) for those Japanese words in katakana which cannot be found in JMDict, convert them into their Roman spelling. Through this we get the corresponding English IDs for these Japanese IDs. Then we use the path-similarity based strategy to align these ontologies.

3 General comments

3.1 Comments on the results

From the results we can see that RiMOM can take advantage of all kinds of information on the ontologies to achieve high performance. The linguistic information is especially important for RiMOM. The structure information and the instance information make a good improvement on the results. When the linguistic information is not available (for example, when the labels of entities are meaningless), the structure information and other information are very important.

Strategy selection is effective in the alignment process. With strategy selection, RiMOM can avoid some noise produced by some strategies when the information these strategies rely on is not adequate. This is a very interesting issue: how to find the best strategy (combination) for a specific matching task. Although we add the MF factor this year compared to last year, it is far from the ideal solution for the strategy selection problem.

We adjust some refinement strategies this year and this change is very helpful in the real ontology matching problem. We also re-implement some of our propagation strategies to make them more efficient so they can be applied on the large scale tasks. With these improvements, RiMOM performs better on large scale data sets such as anatomy and fao.

Since the cross-lingual matching tasks are introduced this year, we make a trial on the process of Japanese ontologies and get a fairly good result. We think the cross-lingual task is very important in ontology matching.

3.2 Discussions on the way to improve the proposed system

First of all, we are very eager to improve our strategy selection mechanism. There are two major issues: 1) what are the essential features of an ontology and what are the essential similarity features between ontologies? How should we describe these features? 2) How to do the strategy selection automatically and more effectively based on these features.

Secondly, we will improve the capability of RiMOM to deal with large scale ontologies. Up to now most strategies in RiMOM cannot be applied to large scale ontologies because of memory and time limit. The vlcr task of OAEI 2008 will be a great challenge.

3.3 Comments on the OAEI 2008 test cases

The benchmark test is better defined than OAEI 2007. The data set is very interesting and makes it easy to find the strength and weakness of matching systems. It is very helpful for us to improve our system.

The mldirectory data set is very interesting. It is a very good challenge to deal with the multi-lingual ontology matching tasks.

In the directory data set, however, there may be conflicts in the “chain” hierarchy. That is to say, there are concepts with more than 1 super-concepts and sub-concepts. We think the problem comes from that a folder may have a sub folder with the same name. When extracting the ontologies, the folder and its same-named folder are given the same URI.

4 Conclusion

In this report, we have briefly introduced how we employed RiMOM to obtain the alignment results in OAEI’08 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 make our comments on the results.

Acknowledgments. The work is supported by the National Natural Science Foundation of China (90604025), Chinese National Key Foundation Research and Development Plan (2007CB310803), and IBM SUR project.

References

1. Euzenat, J., Shivaiko. P.: Ontology Matching. Springer-Verlag, Berlin-Heidelberg, 2007.
2. Kalfoglou, Y., Schorlemmer, M.: Ontology Maching: The State of the Art. The Knowledge Engineering Review Journal, 2003.
3. Tang, J., Li, J., Liang, B., Huang, X., Li, Y., and Wang, K.: Using Bayesian Decision for Ontology Alignment. Journal of Web Semantics, Vol(4) 4, pp. 243-262, 2006.
4. http://wordnet.princeton.edu/
5. http://chasen-legacy.sourceforge.jp/
6. http://www.csse.monash.edu.au/~jwb/j_jmdict.html
7. Felzenszwalb, P.F. and Huttenlocher, D.P.: Efficient belief propagation for early vision. International Journal of Computer Vision, Vol. 70, No. 1, October 2006
8. Melnik, S., Garcia-Molina, H. and Rahm, E.: 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