RiMOM Results for OAEI 2016

Yan Zhang, Hailong Jin, Liangming Pan, Juanzi Li

Tsinghua University, Beijing, China.
{z-y14, jinhl, panlm14}@mails.tsinghua.edu.cn
ljz@keg.tsinghua.edu.cn

Abstract. This paper presents the results of RiMOM in the Ontology Alignment Evaluation Initiative (OAEI) 2016. RiMOM participated in all three tracks of Instance Matching this year. In this paper, we first describe the overall framework of our system (RiMOM). Then we detail the techniques used in the framework for instance matching. Last, we give a thorough analysis on our results and discuss some future work on RiMOM.

1 Presentation of the system

With the rapid development of the Semantic Web, knowledge base has become a dominant mechanism to represent the data semantics on the Web. In practice, data is always distributed on heterogeneous data sources. For example, there are a large number of ontological knowledge bases nowadays, such as DBpedia[1]. , YAGO [2, 3], Xlore [4], etc. It is inevitable that the knowledge about the same real-world entity may be stored in different knowledge bases. Therefore, data integration process requires the detection of such heterogeneous instances to ensure the integrity and consistency.

Most recently, it should be noticed that there are many knowledge bases described in different languages. For example, Wikipedia, a well-known public encyclopedia, contains 281 language versions. It is going to be norm that the same real-world entities are described by different language. Thus, there is a growing need to align instances in a cross-lingual environment so that we can share knowledge from all over the world. In consideration of this circumstance, based on previous version of RiMOM[5], we propose an extended version, which provides support for cross-lingual instance matching in a supervised or an unsupervised way.

There are three major techniques in our system, blocking, multi-strategy, machine learning:

1. **Blocking:** We index the instances based on their objects in two knowledge bases respectively, and then select the instances which contain the same keys as candidate instance pairs. We limit the number of pairs to be compared by this step, which significantly improve the efficiency of the system.
2. **Multi-strategy:** We implement several matchers in our instance matching system, we can execute these matchers in parallel and then aggregate the result according to the characteristics of the source ontologies.
3. **Machine learning:**In general, there are some existing alignments. For exmaple, there are a number of cross-lingual links between two different language versions

of Wikipedia. To make full use of these data, we formalize the instance matching as a binary classification problem, and use the reference mappings to train a classifier, which will determine whether an instance pair is equivalent or not.

Faced with challenges in large-scale instance matching, we propose an novel data integration framework RiMOM-2016 (the latest version of RiMOM), which is based on our former ontology and instance matching system RiMOM [5, 6]. The RiMOM-2016 framework is designed for large-scale and cross-lingual instance matching task specially. It presents a novel multi-strategy method to be fit for different kinds of ontology and employs a learning-based approach to get instance alignments in multilingual environments.

1.1 State, purpose, general statement

This section describes the overall framework of RiMOM2016. The overview of the instance matching system is shown in Fig. 1. The system includes seven modules, i.e., *Preprocess*, *Predicate Alignment*, *Mathcher Choosing*, *Candidate Pair Generation*, *Matching Score Calculation*, *Instance Alignment* and *Validation*. The sequences of the process are shown in the Fig. 1. We illustrate the process as follows.

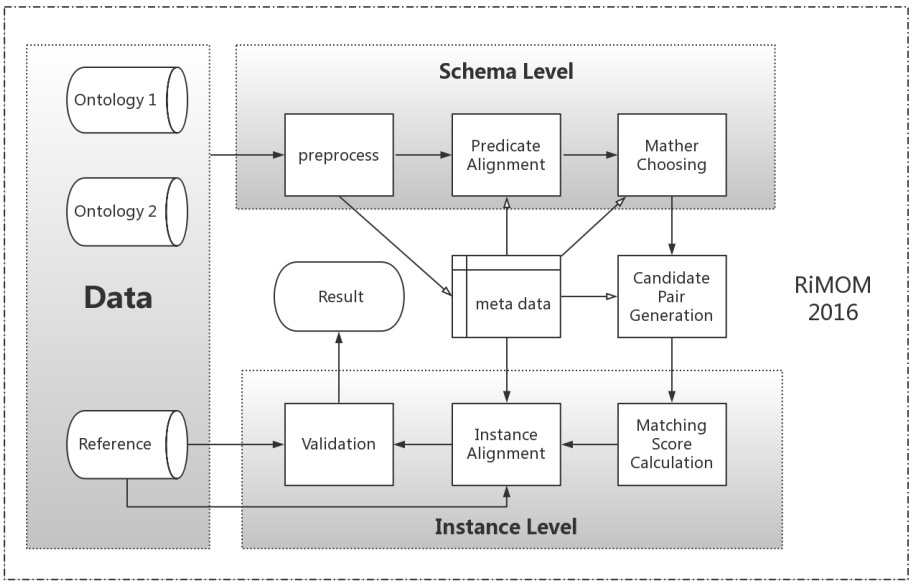

Fig. 1. Framework of RiMOM 2016

1. **Preprocess:** The system begins with *Preprocess*, which loads the ontologies and parameters into system. In the meantime, preprocessor can get some meta data about the two ontologies, which will be used in the later processes, *Predicate alignment* and *Matcher choosing*
2. **Predicate Alignment:** In this process, we will get the alignments of the predicates between the two ontologies.
3. **Matcher choosing:** The system will choose the most suitable one or more matchers according to the meta data of the ontologies.
4. **Candidate Pairs Generation:** In this step, we get candidate pairs when the instances have the same literal objects on some discriminatory predicates.
5. **Matching Score Calculation & Instance Alignment:** This procedure is the most striking difference with the last version of RiMOM. In RiMOM-2016, we get alignments in a supervised or an unsupervised way which depends on whether there exist reference alignments or not. In case of unsupervised method, we calculate similarities between two instances on each property, and then we aggregate these similarities according to the degree of identifying obtained in step 1. On the contrary, we conduct a supervised method when there exist reference alignments. For each instance pairs, we also calculate the similarities as unsupervised way. Then we construct a similarity vector for each pairs and train a logistic regression model [7]. For each candidate instance pair, we use this model to determine whether it is equivalent or not.
6. **Validation:** We will evaluate the alignment result on Precision, Recall and F1-Measure if there is validation data set.

1.2 Specific techniques used

This year we participate in all of three subtasks in the Instance Matching track. We will describe specific techniques in this section.

Data Preprocessing: First, we remove some stop words like "a, of, the", etc. Afterwards, we calculate the TF-IDF values of words in each knowledge base. We also calculate some information of each predicate, in order to obtain the degree of identifying of predicates which will be used in similarity aggregation.

Predicate Alignment: The predicates can express rich semantics, and there exist one-to-one, one-to-many, or many-to-many relationships among these predicates. It is apparent that we should get the alignments of the predicates before we calculate the similarity of instances. In RiMOM-2016, we use an object-based method to align predicates, which is similar with RiMOM-2015 [5].

Blocking: This step aims to pick a relatively small set of candidate pairs from all pairs. Due to the large scale of knowledge bases, it is impossible to calculate matching scores of all instance pairs. In our method, we firstly generate the inverted index on the objects. instance pairs are selected into the candidate set when they have common objects. This method may reduce the recall slightly, but it also reduce the scale of computation significantly.

Multi-Strategy: We implement several matchers in our system, e.g. label-based approach and structure-based approach. In the preprocess step, we will compare the

schema of the two ontologies. If the range of predicates is similar, the label-based approach will play a key role in the matching process. Otherwise, the literal properties are not similar (e.g. the two ontologies are defined in different languages or the intersection of values is really small), label-based approach will not be effective. In this case, we will get some supplementary information (e.g. machine translation, WordNet), or use structure-based appraoch (or use the structure similarity as a feature). In addition, we will use a learning-based method if we have data for training.

Similarity Calculation & Instance Alignment: In OAEI 2016 instance matching track, some of subtasks are defined in the same language, while others use multilingual data sets (e.g. SABINE Task).

Unsupervised method: we use a object-based method to get alignments, it is defined as follows:

$$f_{p_n}(i_1, i_2) = Sim(O_{i_1}^{p_n}, O_{i_2}^{p_n^{'}}) \tag{1}$$

where i_1 and i_2 are instances from two data sets respectively. $O_{i_1}^{p_n}$ represent the object value of instance i_1 on property p_n. $Sim(O_{i_1}^{p_n}, O_{i_2}^{p_n^{'}})$ represent the similarity of object values between these two instances on property p_n and its corresponding property $p_n^{'}$. The computing method of this similarity depends on the data type. For example, we use Levenshtein distance for *type:text* and indicator function for *type:int*.

$$Sim(i_1, i_2) = \omega_1 \times f_{p_1}(i_1, i_2) + \omega_2 \times f_{p_2}(i_1, i_2) + ... + \omega_n \times f_{p_n}(i_1, i_2) \tag{2}$$

For each property p_j, we calculate the similarity according to equation 1 and aggregate them by weights ω_j which indicate the importance of properties.

Supervised method: In equation 2, the weight w_i is determined by meta-data of ontology or manual. Intuitively, it could be improved by a learning-based method if we have some existing alignments. So, basically, we formulate this instance matching problem as a binary classification problem. For a pair of instance i_1 and i_2, the feature vector $\boldsymbol{f} = \{f_{p_i}\}_{i=1}^{n}$. Thus, we can use a sigmoid function to compute the probability that instances i_1 is equivalent with i_2.

$$P(i_1 \equiv i_2) = \frac{1}{1 + e^{\boldsymbol{w} \cdot \boldsymbol{f}(i_1, i_2)}} \tag{3}$$

If $i_1 \equiv i_2$, $P(i_1 \equiv i_2) > 0.5$; otherwise $P(i_1 \equiv i_2) < 0.5$ In this case, the weights $\boldsymbol{w}$ can be determined by the maximum likelihood estimation technique for logistic regression. The assumption in this model is that we can use the machine learning method to determine which property is more important for instance matching problem.

1.3 Link to the system and parameters file

The RiMOM system and configuration files (2016 version) can be found at `https://drive.google.com/file/d/0BzqVVt4Q8YUuaHpseWJOZkI4MnM/view?usp=sharing`.

2 Results

The Instance Matching track contains three tracks and seven subtasks. RiMOM-2016 participate in all of these tracks, and we will present the results and related analysis in this section.

2.1 SABINE Track

There are two subtasks in this track: Inter-linguistic mapping and Data linking. **Table 1** is the result for *Inter-linguistic mapping* task and **Table 2** is for *Data linking* task. Inter-linguistic mapping is a cross-lingual task between English and Italian. As shown in the result, RiMOM preform well in this task. Data linking task requires participants to link the entity to DBpedia, and RiMOM get high Recall but low Precision in this task.

Tool	Precision	Recall	F-measure
LogMapIm	0.012	0.016	0.014
AML	0.919	0.916	0.917
LogMapLite	0.358	0.153	0.214
RiMOM	0.955	0.932	0.943

Table 1. The result for Inter-linguistic mapping

Tool	Precision	Recall	F-measure
LogMapIm	NaN	0.000	NaN
AML	0.926	0.855	0.889
LogMapLite	NaN	0.000	NaN
RiMOM	0.424	0.917	0.580

Table 2. The result for Data linking

2.2 SYNTHETIC Track

There are two subtasks in this track: UOBM and SPIMBENCH. Each subtask contains two data set in different size: *sandbox* is small data set while *mainbox* is a large one. **Table 3 4 5 6** show the final results in this track. We think RiMOM produce satisfactory results in all of the subtasks.

Tool	Precision	Recall	F-measure
LogMapIm	0.701	0.207	0.320
AML	0.785	0.577	0.665
RiMOM	0.771	0.877	0.821

Table 3. The result for UOBM sandbox

Tool	Precision	Recall	F-measure
LogMapIm	0.625	0.023	0.044
AML	0.509	0.515	0.512
RiMOM	0.443	0.516	0.477

Table 4. The result for UOBM mainbox

2.3 DOREMUS Track

This track contains three subtasks: *9-heterogeneities*, *4-heterogeneities*, and *False Positive Trap*. **Table. 7** shows the final result in this track.

Tool	Precision	Recall	F-measure
LogMapIm	0.958	0.766	0.851
AML	0.907	0.749	0.820
RiMOM	0.984	1.000	0.992

Table 5. The result for SPIMBENCH sandbox

Tool	Precision	Recall	F-measure
LogMapIm	0.981	0.695	0.814
AML	0.900	0.747	0.816
RiMOM	0.991	1.000	0.995

Table 6. The result for SPIMBENCH mainbox

Sub-task	Precision	Recall	F-measure
9-heterogeneities	0.813	0.813	0.813
4-heterogeneities	0.746	0.746	0.746
False Positive Trap	0.707	0.707	0.707

Table 7. The result for DOREMUS Track

2.4 Discussions on the way to improve the proposed system

Our system can only align two ontologies at a time, and we think it will be a significant improvement if we can develop a system which is able to align several ontologies simultaneously. In addition, in cross-lingual environment, our system still rely on the machine translation. In this case, we hope to develop a method which is language-independent.

3 Conclusion and future work

In this paper, we present the system of RiMOM in OAEI 2016 Campaign. We participate all of the three tracks in instance matching track this year. We described specific techniques we used in the task. In our project, we design a new framework to align instances in different languages. The results turn out that our method is effective.

In the future, we will make great efforts to improve our system continuously.

4 Acknowledgement

The work is supported by 973 Program (No.2014CB340504), NSFC-ANR (No.61261130588),and NSFC key project(No.61533018), Tsinghua University Initiative Scientific Research Program (No.20131089256) and THU-NUS NExT Co-Lab.

References

1. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia - A crystallization point for the web of data. J. Web Sem. **7**(3) (2009) 154–165
2. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: YAGO2: A spatially and temporally enhanced knowledge base from wikipedia. Artif. Intell. **194** (2013) 28–61
3. Mahdisoltani, F., Biega, J., Suchanek, F.: Yago3: A knowledge base from multilingual wikipedias. In: 7th Biennial Conference on Innovative Data Systems Research, CIDR Conference (2014)
4. Wang, Z., Li, J., Wang, Z., Li, S., Li, M., Zhang, D., Shi, Y., Liu, Y., Zhang, P., Tang, J.: Xlore: A large-scale english-chinese bilingual knowledge graph. In: Proceedings of the ISWC 2013 Posters & Demonstrations Track, Sydney, Australia, October 23, 2013. (2013) 121–124
5. Zhang, Y., Li, J.: Rimom results for oaei 2015. Ontology Matching (2015) 185
6. Li, J., Tang, J., Li, Y., Luo, Q.: Rimom: A dynamic multistrategy ontology alignment framework. IEEE Trans. Knowl. Data Eng. **21**(8) (2009) 1218–1232
7. Hosmer, D.W., Lemeshow, S.: Introduction to the logistic regression model. Applied Logistic Regression, Second Edition (2000) 1–30