InsMT / InsMTL Results for OAEI 2014 Instance Matching

Abderrahmane Khiat[1], Moussa Benaissa[1]

[1] LITIO Lab, University of Oran, BP 1524 El-Mnaouar Oran, Algeria
abderrahmane_khiat@yahoo.com
moussabenaissa@yahoo.fr

Abstract. InsMT and InsMTL are automatic instance-based ontology alignment systems which (a) annotate instances as first step. In the second step, the InsMT system (b) applies different terminological matchers with a local filter on these annotated instances. Contrary to InsMT, the InsMTL system (b) matches the annotated instances not only at terminological level but also at linguistic level. For the first version of our systems and the first participation at OAEI 2014 evaluation campaign, the results are good in terms of recall but they are not in terms of F-measure.

1 Presentation of the system

1.1 State, purpose, general statement

The instance matching aims to identify similar instances among different ontologies. The systems InsMT (**Ins**tance **M**atching at **T**erminological level) and InsMTL (**Ins**tance **M**atching at **T**erminological and **L**inguistic level) are realized for this purpose. InsMT and InsMTL are automatic instance-based ontology alignment that generates as output an alignment which that contains all the semantic correspondences found between the instances of different concepts of the two ontologies to be aligned.

The InsMT and InsMTL systems annotate the instances as first step with concept and property names.

As second step InsMT uses various string-based matching algorithms i.e. terminological level, these similarities calculated by each algorithm are represented in matrix. InsMT applied a local filter on each matrix, and combines these new similarities with average aggregation method.

Contrary to InsMT, InsMTL system calculates similarities between annotated instances not only at terminological level but also at linguistic level. InsMTL combines the similarities calculated by the various string-based matching algorithms at terminological level, with similarities calculated using an external resource WordNet i.e. at linguistic level. The next step consists in combining the similarities by gives the priority to linguistic matcher otherwise we have used an average aggregation method.

Finally both systems applied a filter in order to select the semantic correspondences between instances of different ontologies.

The details of each step of InsMT and InsMTL systems are described in the following section.

1.2 Specific techniques used

The process of InsMT and InsMTL systems consists in the following two successive steps: 1) Annotation and Calculation of Similarities and 2) Combination and Extraction of Alignment.

A. InsMT system

1.2.1 Step 1: Annotation and Calculation of Similarities

1.2.1.1 Phase 1: Extraction of Entities of the Ontologies

In this phase, our system takes as input the two ontologies to be aligned and extract their instances.

1.2.1.2 Phase 2: Annotation of Instances

In this phase, our system annotates in this second step the instances with the name and label of the concept also with property name. The purpose of this annotation is to enrich the instances with terminological information. This step is very import especially when instances do contain terminological information.

1.2.1.3 Phase 3: The Applied Matchers

In this phase, our system calculates the similarities between instances, annotated in previous phase, using various string-based matching algorithms. More precisely the different string-based matching algorithms used are: levenshtein-distance, Jaro, SLIM-Winkler. The calculations of similarities by each string matching algorithm are represented in matrix.

1.2.2 Step 2: Combination and Extraction of Alignment

1.2.2.1 Phase 1: Local Filter

In this first phase of the second step, our system applies a local filter on each matrix i.e. we choose for each string-based matching algorithm a threshold to realize a filter. We consider that: the similarities which are less than the threshold are set to 0. Our intuition behind this local filter is that the similarities which are less than the threshold can influence the strategy of the average aggregation.

1.2.2.2 Phase 2: Aggregation of Similarities

In this phase, our system combines the similarities of each matrix (after we have applied a local filter) using the average aggregation method and the result of the aggregation is represented in a matrix.

1.2.2.3 Phase 3: Global Filter and Identification of Alignment

In this final phase, our system applies a second filter on the combined matrix (result of the previous step) in order to select the correspondences found using the maximum strategy with a threshold.

B. InsMTL system

We mention in this section the difference between InsMT and InsMTL system.

First, we have added another matcher at linguistic level for InsMTL system in second phase “The applied Matchers”, we have used an external dictionary WordNet.

In second step, InsMT does not apply a local filter (phase 1.2.2.1), the similarities calculated by each matcher are represented in matrix without a local filter.

In the phase “Aggregation of Similarities”, InsMTL system gives priority to WordNet i.e. if the similarity value calculated using WordNet is greater than the similarity value calculated using string matching algorithms, the similarity value of the matrix combined is equal to the similarity calculated using WordNet, else we use the average aggregation method. The result of the aggregation is represented in a matrix.

1.3 Adaptations made for the evaluation

We do not have made any specific adaptation for the first version of InsMT and InsMTL, for OAEI 2014 evaluation campaign.

1.4 Link to the system and parameters file

The first version of InsMT and InsMTL systems submitted to OAEI 2014 can be downloaded from seal-project at http://www.seals-project.eu/.

1.5 Link to the set of provided alignments (in align format)

The results of InsMT and InsMTL systems can be downloaded from seal-project at http://www.seals-project.eu/.

2 Results

In this section, we present the results obtained by running InsMT and InsMTL on instance matching track of OAEI 2014 evaluation campaign.

2.1 Instance Matching

The instance matching track aims at evaluating tools able to identify similar instances among different RDF and OWL ontologies. Our both systems annotate the instances with concept and property names as a first step. Then as second step, the InsMT system uses various string-based matching algorithms on annotated instances in order to find correspondences between them and the InsMTL system use another matcher at linguistic level in order to select semantic correspondences between instances of different concepts.

The table 1 and table 2 below present the results obtained by running InsMT and InsMTL on the instance matching track of OAEI campaign 2014.

2.2.1 Identity Recognition Task

The goal of the id-rec task is to determine when two OWL instances describe the same real-world entity.

Identity Recognition Task	Precision	Recall	F-measure
InsMT	0.0008	0.7785	0.0015
InsMTL	0.0008	0.7785	0.0015

Table 1. The results of InsMT and InsMTL on the Identity Matching track of OAEI 2014.

2.2.2 Similarity Recognition Task

The goal of the sim-rec task is to evaluate the degree of similarity between two OWL instances, even when the two instances describe different real-world entities.

Identity Recognition Task	F-measure
InsMT	d(InsMT) = 37.03

Table 2. The results of InsMT on the Similarity Matching track of OAEI 2014.

3 General comments

3.1 Comments on the results

This is the first time that our systems participate in instance matching track of the OAEI 2014 evaluation campaign, and our InsMT and InsMTL systems are new on the SEALS Platform. However they provide good result in terms of recall but not good result in terms of F-measure.

3.2 Discussions on the way to improve the proposed system

The InsMT and InsMT are automatic instance-based ontology matching systems designed in order to find the correspondence between instances of different concepts.

The objective behind the implementation of InsMT and InsmTL systems is first to find the best strategy of annotation. The InsMT system applied different strategy of aggregation and filter as we have proposed in section in section 1.2.1.3 (a local filter). Contrary to InsMT, the objective behind the implementation of AOTL system is to discover more new semantic correspondences by adding other matchers. For now, we have used matchers at terminological and linguistic level.

As we have mentioned before InsMT and InsMTL systems use terminological information for annotation and matching, and when these ontologies do not contain this information our two systems fails. Our both systems does not deal with instances of ontologies written in different languages, and we hope in the future add a module to translate them in the same language.

Another point to be discussed is how to make our systems flexible i.e. the choice of thresholds for the various matchers (terminological and linguistic). It is obvious that we cannot set the threshold for all instances, in order to find automatically the correspondences between instances of ontologies to be aligned; because each ontology contain instances and possesses its own specific characteristic.

4 Conclusion

This is the first time that InsMT and InsMTL have participated at SEAL platform and OAEI 2014. The InsMT and InsMTL are instance-based ontology alignment system, and in this year, our both systems have participated in instance matching track of OAEI 2014 evaluation campaign.

Initially AOT and AOTL systems annotate instances with concept and property names. The purpose of this annotation is to enrich the instances with terminological information.

The InsMT system calculates similarities between these annotated instances using various string-based matching algorithms. The similarities (between these annotated instances) calculated by these different matchers are combined using average aggregation after we have applied a local filter on each matrix.

The InsMTL calculates similarities between these annotated instances using the terminological and linguistic matchers. The similarities (between these annotated instances) calculated by these different matchers are combined using average aggregation with the priority to linguistic matcher.

As final step both systems applied a filter on the combined matrix for the selection of semantic correspondences between different instances of different concepts of ontologies.

Finally the results show that our systems provide good results in terms of recall but they are not in terms of F-measure. We envision to select the best aggregation and filtering strategy and add other matchers such as structure-based and reasoning-based matchers.

References

1. J. Euzenat and P. Valtchev, “Similarity-based ontology alignment in owllite,” in Proceedings of ECAI, (2004).
2. J. Euzenat and P. Shvaiko. OntologyMatching. Springer (2007).
3. M. Ehrig. Ontology Alignment Bridging the Semantic Gap. Springer (2007).
4. M. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. Journal of America Statistical Association, 84(406):414-420, (1989).
5. A. Khiat et M. Benaissa: "Nouvelle Approche d’Alignement d’Ontologies à base d’Instances : trasferet des instances par l’inférence", In The Proceeding of International Conference On Artificial Intelligence and Information Technology, ICA2IT 2014, Ouargla, Algeria, (2014).
6. V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady, 10:707-710, (1966).
7. M. Ehrig and Y. Sure, “Ontology mapping - an integrated approach,” in Proceedings of the European Semantic Web Symposium ESWS, (2004).
8. G. A. Miller, R. Beckwith, C. D. Fellbaum, D. Gross and K. Miller. WordNet: An online lexical database. Int. J. Lexicograph. 3, 4, pp. 235–244, (1990).
9. M. A. Rodriguez and M. J. Egenhofer: Determining Semantic Similarity among Entity Classes from Different Ontologies. IEEE Transactions on Knowledge and Data Engineering, vol. 15, issue 2, pp, 442—456, (2003).
10. A. Doan, J. Madhavan, P. Domingos, and A. Halevy, “Learning to map ontologies on the semantic web,” in Proceedings of the International World Wide Web Conference (2003).
11. A. Maedche, B. Motik, N. Silva and R. Volz "Mafra—a mappingframework for distributed ontologies", Springer, Benjamins VR (eds) EKAW, Berlin, vol 2473, pp 235–250, (2002).
12. K. Todorov, P. Geibel, KU. Kuhnberger "Mining concept similarities for heterogeneous ontologies", Springer, Berlin, ICDM, vol 6171. , pp 86–100, (2010).
13. B. Schopman, S. Wang, A. Isaac and S. Schlobach, “Instance-Based Ontology Alignment by Instance Enrichment”, *Journal on Data Semantics*, vol. 1, N° 4, (2012).
14. E. Rahm “Towards large-scale schema and ontology Alignment”, *ReCALL*, (2011).
15. J. Li, J. Tang, Y. Li and Q. Luo, “Rimom: a dynamic multistrategy ontology alignment framework”, *IEEE Trans Knowl*, (2009).