oMAP: Results of the Ontology Alignment Contest*

Umberto Straccia
ISTI-CNR, Via G. Moruzzi 1
56124 Pisa, Italy
straccia@isti.cnr.it

Raphaël Troncy
CWI Amsterdam, P.O. Box 94079
1090 GB Amsterdam, The Netherlands
raphael.troncy@cwi.nl

ABSTRACT

This paper summarizes the results of the *oMAP* system for the 2005 Campaign tests of the Ontology Alignment Evaluation Initiative. First, it describes the system and its main components. The results of the experiments for the three tasks follow. Then, we have a short discussion and interpretation of the results. Finally, we sketch some ideas to improve our system and provide the link to our current results.

1. PRESENTATION OF THE SYSTEM

oMAP [12] is a framework whose goal is to automatically align two OWL ontologies, finding the best mappings (together with their weights) between the entities defined in these ontologies. The final mappings are obtained by using the prediction of different classifiers. For this experiment, we have used terminological and machine learning-based classifiers, plus a new one, based on the structure and the semantics of the OWL axioms.

The *oMAP* implementation allows to align any OWL ontologies, represented in the RDF/XML syntax. Hence, it uses extensively the OWL API [1] and the Alignment API available in JAVA [3].

1.1 State, purpose, general statement

Our approach is inspired by the data exchange problem [4] and borrows from others, like GLUE [2], the idea of using several specialized components for finding the best set of mappings. The framework resumes partially the formalization proposed in [7] and extends the sPLMAP (*Schema Probabilistic Learning Mappings*) system to cope with the ontology alignment problem.

Theoretically, an ontology *mapping* is a tuple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where $\mathbf{S}$ and $\mathbf{T}$ are respectively the source and target ontologies, and Σ is a finite set of *mapping constraints* of the form:

$$\alpha_{i,j}\ T_j \quad \leftarrow \quad S_i$$

where S_i and T_j are respectively the source and target entities. The intended meaning of this rule is that the entity S_i of the source ontology is mapped onto the entity T_j of the target ontology, and the confident measure associated with this mapping is $\alpha_{i,j}$. Note that a source entity may be mapped onto several target entities and conversely. But, we do not require that we have a mapping for every target entity.

Aligning two ontologies in *oMap* consists of three steps:

1. We form a possible Σ, and estimate its quality based on the quality measures for its mapping rules;
2. For each mapping rule $T_j \leftarrow S_i$, we estimate its quality $\alpha_{i,j}$, which also depends on the Σ it belongs to, i.e. $\alpha_{i,j} = w(S_i, T_j, \Sigma)$;
3. As we cannot compute all possible Σ (there are exponentially many) and then choose the best one, we rather build iteratively our final set of mappings Σ using heuristics.

Similar to GLUE [2], we estimate the weight $w(S_i, T_j, \Sigma)$ of a mapping $T_j \leftarrow S_i$ by using different classifiers $CL_1, \ldots, CL_n$. Each classifier CL_k computes a weight $w(S_i, T_j, CL_k)$, which is the classifier's approximation of the rule $T_j \leftarrow S_i$. For each target entity T_j, CL_k provides a rank of the plausible source entities S_{i_k}. Then we rely on a priority list on the classifiers, $CL_1 \prec CL_2 \prec \ldots \prec CL_n$ and proceed as follows: for a given target entity T_j, select the top-ranked mapping of CL_1 if the weight is non-zero. Otherwise, select the top-ranked mapping provided by CL_2 if non-zero, and so on.

In the next section, we briefly present the classifiers that are currently used in our framework. It is worth not-

*This work was carried out during the tenure of an ERCIM fellowship at CNR.

ing that some of them consider the terminological part of the ontologies only, while others are based on their instances (i.e. the values of the individuals). Finally, we end this section by introducing a new classifier that fully uses the structure and the semantics of ontology definitions and axioms.

1.2 Specific techniques used

The terminological classifiers work on the name of the entities (class or property) defined in the ontologies. In OWL, each resource is identified by a URI, and can have some annotation properties attached. Among others, the `rdfs:label` property may be used to provide a human-readable version of a resource's name. Furthermore, multilingual labels are supported using the language tagging facility of RDF literals. In the following, we consider that the name of an entity is given by the value of the `rdfs:label` property or by the URI fragment if this property is not specified. The typical terminological classifiers we used in *oMAP* compare the name of the entities, their stem (using the Porter stemming algorithm [9]), compute some similarity measures between the entity names (once downcased) such that the Levenshtein distance[6] (or edit distance), or compute similarity measure between the entity names using the WordNet®[1] relational dictionary.

Additionally, an ontology often contains some individuals. It is then possible to use machine learning-based classifiers to predict the weight of a mapping between two entities. The instances of an OWL ontology can be gathered using the following rules: we consider (*i*) the label for the named individuals, (*ii*) the data value for the datatype properties and (*iii*) the type for the anonymous individuals and the range of the object properties. For example, using the abstract syntax of [5], let us consider the following individuals :

Individual (x_1 type `(Conference)`
  value `(label "Int Conf on Knowledge Capture")`
  value `(location` x_2`))`
Individual (x_2 type `(Address)`
  value `(city "Banff")` value `(country "Canada"))`

Then, the text gathered u_1 for the named individual x_1 will be (`"Int Conf on Knowledge Capture", "Address"`) and u_2 for the anonymous individual x_2 (`"Address", "Banff", "Canada"`). Typical and well-known classifiers used in machine learning such as Naive Bayes and kNN [11] have then been implemented in *oMAP* using these data.

Finally, a new classifier is able to use the semantics of the OWL definitions while being guided by their syntax. This *structural classifier* is fully described in [12]. It is used in the framework *a posteriori*. Indeed, we rely on the classifier preference relation $CL_{Name} \prec CL_{Stem} \prec CL_{EditDistance} \prec CL_{NaiveBayes}$. According to this preference relation, a set Σ' of mappings is determined. This set is given as input to the structural classifier. Then the structural classifier tries out all alternative ways to extend Σ' by adding some $T_j \leftarrow S_i$ if no mapping related to T_j is present in Σ'.

1.3 Adaptations made for the contest

All the classifiers detailed previously have been implemented to be compatible with the alignment API[2], thus easing their chaining. Therefore, our *oMAP* framework benefits from all the evaluation facilities for comparing our approach with other methods.

For the purpose of this contest, all our classifiers have been tested alone and then combined. We have then made no specific adaptations since we are still investigating how the classifiers should be combined to improve the overall quality of *oMAP*.

2. RESULTS

The tests proposed by the 2005 campaign of the Ontology Alignment Evaluation Initiative is composed of three tasks. Below, we describe the results of the *oMAP* system for these three tasks as well as the problems we have encountered.

2.1 Task1: benchmarks

The *benchmarks tests* are systematic benchmarks series produced for identifying the areas in which each alignment algorithm is strong and weak. Taking back the tests of the 2004 contest [13] and extending them, there are based on one particular ontology dedicated to the very narrow domain of bibliography and a number of alternative ontologies of the same domain for which alignments are provided. The full table results for this task is given in the section 6.3.

The overall score of *oMAP* for this task is quite good (see the table below).

Tests	Prec.	Rec.
1xx	0.96	1.00
2xx	0.80	0.63
3xx	0.93	0.64
H-Mean	0.83	0.66

However, *oMAP* has poor performance for the tests 25x and very bad performance for the tests 26x. Actually, the terminological and machine-learning based classifiers give wrong input to our structural classifier, since most of the data used in these classifiers have been changed in these tests. The structural classifier is then not able to counterbalance this effect and give also wrong alignments. It is the typical case where the

[1]WordNet: `http://wordnet.princeton.edu/`.

[2]`http://co4.inrialpes.fr/align/`.

other classifiers should be turn off and the structural classifier should work alone.

2.2 Task2: directory

The *directory real world case* consists of aligning web sites directory. It is more than two thousand elementary tests. These tests are blind in the sense that the expected alignments are not known in advance. *oMAP* success to compute the alignments for all of them in a total time of about 11 minutes on a normal PC laptop.

2.3 Task3: anatomy

The *anatomy real world case* covers the domain of body anatomy and consists of two big ontologies with an approximate size of several 10k classes and several dozen of relations. This test was clearly the hardest one and our *oMAP* system has not been able to begin the computation of the alignment.

The main problem is the size of the FMA ontology since the XML parser cannot load the full ontology and crash for *out of memory* problem. However, we notice also that this ontology contains some small mistakes:

- the entities, such that `&xsd; &rdfs; ...` , are not defined;
- a datatype property contains an error since the value of its `rdf:ID` attribute contains a space which is forbidden. The correct definition of this property should be:

```
<owl:DatatypeProperty
    rdf:ID="has_inherent_3-D_shape"
    rdfs:label="has inherent 3-D shape">
        <rdfs:domain rdf:resource="&rdfs;Resource"/>
        <rdfs:range rdf:resource="&xsd;boolean"/>
</owl:DatatypeProperty>
```

Once these mistakes have been corrected, the FMA ontology can be validated with an RDF parser, but the parser included in the alignment API is not able to deal with, thus preventing the beginning of the alignment computation by *oMAP*.

3. GENERAL COMMENTS

3.1 Comments on the results

As we have seen in the previous section, the *oMAP* framework is based on numerous classifiers. Each of them try to predict some mappings between the ontology entities and these predictions are then combined. Some classifiers are strongly based on the labels attached to the entities (terminological). Therefore, they performed especially well when labels were preserved. The machine learning-based classifiers can use the individuals of the ontologies if they contain. The main improvement of our approach is then the structural classifier which is able to align two ontologies solely on their semantics, and without the presence of individuals or even labels.

Finally, the combination of all these classifiers rely on many different features and thus balance the influence of individual features. This mixed approach tend to success on every case (either if the labels are preserved or not) even if we dispose yet of a large progression margin.

The main weakness of *oMAP* is clearly its computation time. Like we have seen previously, our approach begins to form some possible Σ sets, for evaluating the weight of each mapping rules they contain. The generation of *all* possible Σ sets becomes quickly a critical issue since this number can be huge (exponentially many) [12]. We have addressed this problem by implementing some approximation. The most efficient for reducing the space search is a local maximum heuristic. When forming a Σ set, we consider firstly a class from the first ontology, and gather all the entities (classes and properties) involved in its closure definition. We do the same for each classes of the second ontology and we evaluate all these small Σ sets for retaining the best one. We iterate this process over all the classes. Additional criteria allow us to guarantee the convergence of our approach (i.e. the order of the classes considered has no significance).

3.2 Discussions on the way to improve the proposed system

As future work, we see some appealing points. Additional classifiers using more terminological resources can be included in the framework, and are currently under implementation, while the effectiveness of the machine learning part could be improved using other measures like the KL-distance. While to fit new classifiers into our model is straightforward theoretically, practically finding out the most appropriate one or a combination of them is quite more difficult. In the future, more variants should then be developed and evaluated to improve the overall quality of *oMAP*. Furthermore, the appropriateness of each classifier could be learned via regression.

3.3 Comments on the test cases

It is always difficult to create good test cases. The benchmarks tests should cover the widest range of discrepancies occurring when having two ontologies. For the cases where a lot of features were explicitly changed at the same time, *oMAP* is clearly less good, but such a mess is unlikely to occur in real cases. On the contrary, in the real world scenario, as presented by the last four test cases, *oMAP* performs quite good.

4. CONCLUSION

As the number of Semantic Web applications is growing rapidly, many individual ontologies are created. The development of automated tools for ontology alignment will be of crucial importance. In this paper, we have presented the results for the 2005 campaign of the Ontology Alignment Evaluation Initiative of our formal framework for ontology Matching, which for ease we call *oMap*. oMap uses different classifiers to estimate the quality of a mapping. Novel is the classifier which uses the structure of the OWL constructs and thus the semantics of the entities defined in the ontologies. We have implemented the whole framework and we continue to evaluate it on independent benchmark tests such that the ones provided by this contest.

5. REFERENCES

[1] Bechhofer, S., Volz, R., Lord, P.W.: Cooking the Semantic Web with the OWL API. In *2nd International Semantic Web Conference (ISWC'03)*, Sanibel Island, Florida, USA, 2003, 659–675.

[2] Doan, A., Madhavan, J., Dhamankar, R., Domingos, P., Halevy, A.: Learning to Match Ontologies on the Semantic Web. The VLDB Journal, **12**(4), 2003, 303–319.

[3] Euzenat, J.: An API for ontology alignment. In *3rd International Semantic Web Conference (ISWC'04)*, Hiroshima, Japan, 2004, 698–712.

[4] Fagin, R., Kolaitis, P.G., Miler, R.J., Popa, L.: Data Exchange: Semantics and Query Answering. In *9th International Conference on Database Theory (ICDT'03)*, Sienna, Italy, 2003, 207–224.

[5] Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From SHIQ and RDF to OWL: The making of a web ontology language. Journal of Web Semantics, **1**(1), 2003, 7–26.

[6] Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Doklady Akademii Nauk SSSR, **163**(4):845–848, 1965 (Russian). English translation in Soviet Physics Doklady, **10**(8):707–710, 1966.

[7] Nottelmann, H., Straccia, U.: sPLMap: A probabilistic approach to schema matching. In *27th European Conference on Information Retrieval (ECIR'05)*, Santiago de Compostela, Spain, 2005, 81–95.

[8] OWL, Web Ontology Language Reference Version 1.0. W3C Recommendation, 10 February 2004. `http://www.w3.org/TR/owl-ref/`

[9] Porter, M.F.: An algorithm for suffix stripping. Program, **14**(3), 1980, 130–137.

[10] RDF, Ressource Description Framework Primer W3C Recommendation, 10 February 2004. `http://www.w3.org/TR/rdf-primer/`

[11] Sebastiani, F.: Machine learning in automated text categorization. ACM Comuting Surveys, **34**(1), 2002, 1–47.

[12] Straccia, U., Troncy, R.: oMAP: Combining Classifiers for Aligning Automatically OWL Ontologies. In *6th International Conference on Web Information Systems Engineering (WISE'05)*, New York City, New York, USA, 2005.

[13] Sure, Y., Corcho, O., Euzenat, J., Hughes, T. editors. *3rd International Workshop on Evaluation of Ontology-based Tools (EON'04)*, Hiroshima, Japan, 2004.

6. RAW RESULTS

6.1 Link to the system and parameters file

The *oMAP* system is available at: `http://homepages.cwi.nl/~troncy/`.

It can be run with the command:

```
java -jar omap.jar -i %method% -r %renderer%
     -o %resultFile% %sourceOnto% %targetOnto%
```

where:

- *method* is: `it.cnr.isti.OMapAlignment`;
- *renderer* is: `fr.inrialpes.exmo.align.impl.renderer.RDFRendererVisitor2`;
- *resultFile* is the name of the result file;
- *sourceOnto* and *targetOnto* are the absolute URIs of the source and the target ontologies to align.

6.2 Link to the set of provided alignments

All the alignments computed for this campaign are available in the alignment format [3] at: `http://homepages.cwi.nl/~troncy/OAEI/`.

6.3 Matrix of results

#	Name	Prec.	Rec.	Time
101	Reference alignment	0.96	1.00	20ms
102	Irrelevant ontology	0.00	NaN	
103	Language generalization	0.96	1.00	20ms
104	Language restriction	0.96	1.00	20ms
201	No names	0.88	0.38	20ms
202	No names, no comments	0.85	0.24	20ms
203	No comments	0.96	1.00	
204	Naming conventions	0.95	0.89	40ms
205	Synonyms	0.81	0.63	40ms
206	Translation	0.89	0.49	40ms
207		0.89	0.49	40ms
208		0.96	0.90	30ms
209		0.73	0.54	40ms
210		0.90	0.39	40ms
221	No specialisation	0.96	1.00	20ms
222	Flatenned hierarchy	0.96	1.00	20ms
223	Expanded hierarchy	0.96	1.00	20ms
224	No instance	0.96	1.00	20ms
225	No restrictions	0.96	1.00	20ms
228	No properties	0.92	1.00	20ms
230	Flattened classes	0.91	1.00	20ms
231	Expanded classes	0.96	1.00	20ms
232		0.96	1.00	20ms
233		0.92	1.00	20ms
236		0.92	1.00	20ms
237		0.95	1.00	20ms
238		0.96	1.00	20ms
239		0.85	1.00	20ms
240		0.87	1.00	20ms
241		0.92	1.00	20ms
246		0.85	1.00	20ms
247		0.87	1.00	20ms
248		0.85	0.24	50ms
249		0.85	0.23	50ms
250		0.05	0.06	50ms
251		0.85	0.25	50ms
252		0.85	0.24	50ms
253		0.85	0.23	50ms
254		0.06	0.06	50ms
257		0.00	0.00	50ms
258		0.85	0.25	50ms
259		0.85	0.24	50ms
261		0.03	0.03	50ms
262		0.00	0.00	50ms
265		0.00	0.00	50ms
266		0.00	0.00	50ms
301	Real: BibTeX/MIT	0.94	0.25	40ms
302	Real: BibTeX/UMBC	1.00	0.58	40ms
303	Real: Karlsruhe	0.90	0.79	40ms
304	Real: INRIA	0.91	0.91	40ms