X-SOM Results for OAEI 2007

Carlo Curino, Giorgio Orsi, and Letizia Tanca
{curino,orsi,tanca}@elet.polimi.it

Politecnico di Milano
Dipartimento di Elettronica e Informazione
Via Ponzio 34/5, Milano (Italy)

Abstract. This paper summarizes the results of the X-SOM tool in the OAEI 2007 campaign. X-SOM is an extensible ontology mapper that combines various matching algorithms by means of a feed-forward neural network. X-SOM exploits logical reasoning and local heuristics to improve the quality of mappings while guaranteeing their consistency.

1 Presentation of the system

Nowadays, the spreading of data intensive and community-centered web-applications has multiplied the number of available datasources accessible through the Internet. In order to effectively query and integrate this information, a shared formalism should be used, at least as a means to mediate the access to datasources. In many situations, ontologies [8] have demonstrated , to be a suitable formalism for evenly representing the content of heterogeneous datasources [15], with a well-defined semantics. In principle, it is possible to extract an ontology from a datasource, and then integrate its information content with that of other datasources, by relating their respective ontologies.

Ontology mapping is then defined as *the process of bringing two or more ontologies into mutual agreement, by relating their similar concepts and roles by means of alignment relationships*. Generally speaking, the mapping process aims at providing a *unified, consistent and coherent view over multiple conceptualizations of one or more domains of interest.*

In this paper, we briefly describe our ontology mapping tool, X-SOM [5] (eXtensible Smart Ontology Mapper), summarizing the performance obtained against the OAEI 2007 test cases.

The architecture of the X-SOM Ontology Mapper is composed by three subsystems: *Matching*, *Mapping* and *Inconsistency Resolution*.

The Matching Subsystem is constituted by an extensible set of matching modules, each of which implements a matching technique that may be invoked by the mapper according to a configurable *matching strategy*; this strategy defines also the way the matching values are combined. Each module receives as input two ontologies and returns a set of matchings, along with a similarity degree, between *homogeneous* resources (i.e., concepts with concepts, roles with roles and individuals with individuals); the produced structure is called *similarity map*.

All similarity maps produced by the Matching Subsystem are collected by the Mapping Subsystem; the various proposals are then combined by means of a feed-forward neural network in order to produce an aggregated similarity degree, starting from the single similarities computed by each module of the Matching Subsystem. Given these aggregate matching values, the Mapping Subsystem computes a set of *candidate mappings* by applying, to the set of matchings, a pair of configurable threshold values. The first threshold is called *discard threshold*; the matchings with a similarity degree lower than it are discarded a-priori. The second threshold is called *accept threshold*, and the matchings with a similarity degree greater than it are accepted as candidate mappings. The remaining matchings, whose similarity is between the two thresholds, are considered as uncertain and manually evaluated by the user.

Mapping two ontologies might produce inconsistencies [12]; for this reason, the set of candidate mappings computed by the Mapping Subsystem is handed to the Inconsistency Resolution Subsystem, responsible for guaranteeing mappings consistency. Moreover, the X-SOM consistency-checking process can be instructed to preserve the semantics of the original ontologies, in terms of concept definitions and relationships among them. The so-obtained mappings capture the *consensual knowledge* about the domain, i.e., that information which represents an added value for the system, without changing the semantics of the input ontologies and, in turn, without incurring in the need to adapt the applications built upon them.

Ontologies are often published on the Web and not accessible for modifications. For this reason, and to preserve the original representations, X-SOM mappings are stored in a separate ontology called *mapping ontology*. This ontology acts as a "bridge" between the mapped ontologies and can be used to access the global model constituted by source ontologies connected through the mappings. If needed, it is possible to store in the bridge ontology also the concept definitions needed to disambiguate some terms or to solve particular inconsistencies.

X-SOM generates subsumption and equivalence mappings between pairs of resources; they are expressed by means of RDFS and OWLS constructs, in order to maintain the highest interoperability of mapping definitions.

1.1 State, purpose, general statement

X-SOM has been designed to automatically discover useful relationships among ontological representations with the purpose of enabling ontology-based data integration and tailoring [6]. The theoretical framework used in this work is that of DL ontologies; however, the X-SOM approach is very flexible and we believe that it is possible to extend it to other ontology languages, and even to other data models such as XML and the relational model.

X-SOM is part of a wider research project named Context-ADDICT (Context-Aware Data Design, Integration, Customization and Tailoring) [1], which aims at the definition of a complete framework able to support mobile users through the dynamic hooking and integration of new, heterogeneous information sources, until a suitable, contextualized portion of the available data is delivered on their devices, in a structured and offhanded way. The whole process is widely based on ontological representations of both the application domain and datasources; this naturally leads to an ontology mapping process that should be as much automatic as possible.

1.2 Specific techniques used

In this section we describe, in more detail, the three subsystems that constitute the X-SOM architecture.
The Matching Subsystem has been designed to be extensible, to allow easy integration of future matching modules. Since this architecture makes experimenting new modules very easy, X-SOM can also be used as a framework for evaluating matching techniques. X-SOM's matching modules can be roughly classified into three families:

- *language-based:* The modules belonging to this family of algorithms compare resources by analyzing their names, labels and comments. They consider both the lexical and linguistic features as terms of comparison. The lexical modules currently implemented are: the *Jaro module*, based on Jaro String Similarity [4] and the *Levenshtein module* based on the Levenshtein string distance; To exploit linguistic similarities, we implemented a *WordNet module* that uses the WordNet [13] thesaurus, computing some distance measures like the *Leacock-Chodorow* [11].
- *structure-based:* These modules are used to compare the structures of the resources' neighborhoods. In X-SOM, we have implemented a modified version of the GMO (Graph Matching for Ontologies) algorithm [9], used to find structural similarity in ontological representations. Since the GMO algorithm is quite expensive in terms of required computational resources, we implemented a bounded-path matcher called *Walk* that reaches lower performance while requiring less resources.
- *semantics-based:* The modules belonging to this family implement algorithms that use background, contextual and prior knowledge to compute the similarity degree between two resources. At the moment, only a Google-based algorithm, described in [3], is implemented.

The Mapping Subsystem receives as input the set of similarity maps computed by the modules of the Matching Subsystem, and produces a set of candidate mappings to be verified by the Inconsistency Resolution Subsystem.
The most challenging issue is how to aggregate all the contributions coming from the various matching modules. In our setting, the problem has been modeled as the estimation of an optimal aggregation function $y = W(\mathbf{X})$ where each component $x_\imath \in \mathbf{X}$ is the matching degree given by the $\imath^{th}$ module of the schedule with respect to a pair of resources, and y is the computed aggregate similarity.
The Mapping Subsystem is as extensible as the Matching Subsystem previously described; it allows to add new aggregation algorithms to X-SOM, by implementing a simple interface.
At the current development state of the prototype, the most effective aggregation algorithm implemented uses a three-layer, feed-forward neural network. The learning algorithm used is a standard back-propagation algorithm with cross-validation; the values for the moment and the learning-rate have been set after empirical evaluation (i.e., over 50.000 runs of the tool). Notice that the task of determining a good aggregation function is, in general, very complicated, since it is not possible to imagine a unique aggregation function that is suitable for every possible alignment situation. Even by supposing a trivial situation where the W function is approximated with a linear function (e.g., a weighted mean), the process of determining the weight of each module implies that the

user knows in advance how reliable the various techniques are.
Another interesting aspect is how to build a suitable training set for the neural network. In X-SOM, the training set is generated from a manually-aligned pair of ontologies called *reference alignment*; correct mappings generate a sample with desiderata equal to 1.0, while the others will be set to zero. Moreover, a *cleaning process* removes: duplicate samples (i.e., similar inputs and same desiderata), conflicting samples (i.e., same inputs but contradictory desiderata) and linearly dependent samples. In the situation of conflicting samples, only the ones with desiderata equal to 1.0 are kept and the reason resides in the way the desiderata are obtained. To determine if, to a set of inputs, should correspond a positive outcome, the trainer looks at the reference alignment. If the given set of inputs is generated by a correct alignment, the outcome is positive (i.e., 1.0) else, it is set to zero. When two conflicting samples are found, the trainer assumes that the one with positive outcome is correct, while the other is discarded.
It is possible that, in certain situations, a module be not able to produce a similarity degree for a given pair of resources; in this case, the value is approximated by means of an average over the similarity degrees generated by the other modules belonging to the same family.
Once the neural network has produced the aggregate similarity values, X-SOM filters them by means of two configurable thresholds: *accept* and *discard*. These thresholds also determine the level of automation of the tool, called *behavior*, which can be: *Fully-automatic*, *Conservative* or *Human-intensive*. When X-SOM acts with one of the last two behaviors (i.e., supervised behaviors), it is possible to involve the user in deciding what matchings should be accepted. In particular, with the conservative behavior, all the mapping proposals with a similarity degree between the discard and accept thresholds are submitted to the user to be evaluated. When the user does not agree with a X-SOM proposal about a pair of resources, the network trainer performs additional training steps until the result of the network agrees with the user, thus allowing fine-tuning of the network's biases. The human-intensive behavior is very similar to the previous one, only it does not discard any mapping a-priori, leaving to the user the freedom to explore all the mappings with a similarity degree under the accept threshold.

The Inconsistency Resolution Subsystem takes as input the candidate mappings from the Mapping Subsystem and produces a set of mappings, in which at least all the logical inconsistencies have been solved. Since the input ontologies are supposed to be consistent, consistency resolution is reduced to identifying those mappings that introduce a contradiction into the final model. This problem is faced in X-SOM at two different levels: *consistency check* and what we have called *semantic coherence check*.
Consistency check locates those mappings that introduce a logical contradiction in the original ontologies. X-SOM uses an extended tableau algorithm to identify the set of mappings responsible for inconsistency and uses a set of heuristic rules, based on the similarity degree, in order to remove those mappings; since the removal of mappings leads to a loss of information, the rules try to preserve as much information as possible, in terms of logical axioms. Also the inconsistency resolution policies are affected by the tool behavior described above. When the tool acts in a supervised behavior, the inconsistent mappings are submitted to the user who selects the correct ones; wrong mappings are then removed automatically. When acting with the fully-automatic be-

havior, X-SOM removes the less probable mappings using the heuristic rules.
By *Semantic coherence check* we mean the process of verifying whether there are mappings that introduce into the model a *semantic incoherence* without introducing a logical contradiction into the T-BOX. To better explain what we mean by semantic coherence, let us introduce the notion of *local entailment*: an entailment $A \sqsubseteq B$, in the global model, is said to be *local to an ontology O* if it involves only resources of *O*. By semantic incoherence we mean the situation in which the alignment relationships enable one or more local entailments that were not enabled within the original ontologies. This, in general, is a desirable behavior for systems that exploit ontologies; however, in certain situations, it is possible to introduce an incoherent assertion without introducing a logical contradiction into the model.
A simple example of semantic incoherence is the emergence of a cycle of subsumptions after a mapping process, which leads to a collapse of the involved concepts into a unique concept. The collapse of two concepts – which were only in a subclass relationship in the original ontologies – changes the semantics of the representation: for this reason, our algorithm removes the mappings responsible for that behavior. We consider a semantic incoherence as a possible symptom of an inconsistency; since we are interested in developing a high precision ontology mapper, we currently adopt a conservative approach that does not allow any change in the semantics of the original ontologies. The main drawback of this approach is that it is possible to lose some useful inferences on the global model.

1.3 Adaptations made for the evaluation

In order to comply with the test-cases proposed in this contest, we made two main adaptations:

- *External resources:* Using the original configuration of X-SOM, external resources (e.g., FOAF definitions) are imported and used in the mapping process. As a result, also the mappings between pairs of external resources are included in the alignment ontology produced by X-SOM. To avoid a wrong computation of performance measures, we artificially removed this kind of mappings from the output of the tool when they were not part of the reference alignment.
- *Properties comparison:* In some reference alignments, datatype properties are compared and aligned with object properties; since this kind of mapping is normally forbidden in X-SOM, we modified the matching algorithms in order to allow this kind of matching.

1.4 Link to the system and parameters file

X-SOM is an open-source project, since it also relies on existing implementations of known matching algorithms. To obtain a working copy of the X-SOM prototype, along with the source code, please send an email to `orsi@elet.polimi.it`.

1.5 Link to the set of provided alignments

http://home.dei.polimi.it/orsi/xsom-oaei07.zip

2 Results

The tests have been made with a configuration that includes:

- The WordNet-based module.
- The GMO structural module without feeding.
- The Jaro syntactical module.

The best results are obtained by aggregating the results by means of the neural network that has been trained on the `animals.owl` ontology available at the I^3CON Initiative website[1]. X-SOM is implemented in Java and relies on the JENA API. The behavior is fully-automatic with an accept threshold set to 60% of similarity and consistency checking turned on. The presence of consistency checking accounts for the somewhat high execution times. In this section we report the performance of X-SOM for the various OAEI tracks. The tests have been made using a Pentium IV 2.1 GHz with 2 GB of RAM. The JVM has been set with a minimum and maximum heap space of 64 MB and 2 GB respectively. The WordNet-based module relies on the JWNL API version 1.3.

2.1 Benchmark

The test cases belonging to this track can be divided into five categories:

Basics (101-104) This family analyzes the ability of a matcher to make simple alignments and to be robust to variations in the OWL dialect. On these, very simple, tests X-SOM obtains an average precision of 99% and an average recall of 98.6%.

Linguistics (201-210) The test cases belonging to this family manipulate resources' names, comments and labels in order to stress the performance of syntactic and lexical matchers. X-SOM performs quite well thanks to the Jaro and WordNet modules; some problems come out when the tool deals with test case 204 since we are not able to recognize acronyms. We have planned to add a pre-processing step in these modules in order to recognize common naming conventions thus normalizing the name of resources also considering possible compound words. This normalization will be kept internal to each module, avoiding any modification in the input ontologies. In this section, X-SOM reaches an average precision of 81.6% and an average recall of 75.4%.

Structure (221-247) These test cases stress the capabilities of the various matching algorithms of finding similar resources in ontologies with different structure. X-SOM performs very well since it reaches an average precision and recall of 99%.

Systematic (248-266) This family combines the previous techniques by removing systematically the structure or by randomizing the names of the resources. This is the hardest part of the benchmark track since X-SOM obtains useful information from the GMO module only. X-SOM obtains an average precision and recall of 26%. The hardest tests cases are 262 and 265 where the result of the matching process is empty because also the GMO module has not enough information to find the similarities.

[1] http://www.atl.external.lmco.com/projects/ontology/i3con.html

Real ontologies (301-304) In these test cases the reference ontology is aligned with real world ontologies describing the same bibliographic domain. These are the most informative tests since these ontologies includes a set of design choices that make the alignment task quite hard. X-SOM reaches an average precision of 94% and an average recall of 67%. The hardest test for X-SOM is 301 since contains many compound words with different naming conventions.

2.2 Anatomy and Food

These tracks have been the most challenging. The problem resides in their dimension that is too big for the current version of X-SOM. In order to perform the alignment, we needed to partition both ontologies using the partition algorithm implemented in the SWOOP [10] ontology editing and debugging framework. However, this procedure was not enough to reduce the ontologies to a manageable dimension; the partition algorithm used on the NCI thesaurus produced a partition with over 3200 classes that cannot be analyzed by our GMO module thus, for this reason, only lexical modules have been used.

2.3 Directory

The main problem with this track is the modularization of the test cases. The small test-cases are too small to exploit the full power of the GMO module, while the comprehensive ones are too big and exhaust the JVM heap space if not partitioned. These limitations lead the GMO module to return poor answers that, in turn, affect the final results.

3 General comments

X-SOM seems to perform quite well on the OAEI test cases; however, the main problems are represented by the aggregation function and large ontologies processing.

We recall from Section 1.2 that the X-SOM neural network is trained only once and then used for all the proposed tracks. Previous tests, performed using ontology pairs describing different domains, have shown that the learned aggregation function is substantially independent of the domain, but strongly dependent on the ontology design technique [2]. This means that, if the neural network is trained on a pair of ontologies with a rich structure, the learning algorithm will probably keep the results of the structural modules in high consideration since, in general, they are helpful for finding the correct alignments. If the same function is then used in a mapping task concerning ontologies that lack of structure, the poor results generated by the structural modules will affect the final results, lowering the whole performance of the tool. An even better performance would be achieved if the modules' schedule could be changed among different tracks.

The second problem is represented by very large ontologies that require too much memory to be processed with the X-SOM approach. A solution to this problem is the modularization of ontologies and the subsequent mapping of ontology chunks modeling the same portion of the application domain. At the moment we are not planning to address this problem within X-SOM but, for instance, we are considering to resort to modularization algorithms such as that implemented in SWOOP.

3.1 Discussions on the way to improve the proposed system

We are planning to introduce new modules able to extract and reuse the consensual knowledge that emerges in collaborative and social web-applications, in order to disambiguate some mapping situations that generally need user intervention. We are currently exploring other machine-learning techniques for the matchings combination task [7], in particular *white-box* techniques like *decision-tree learning*. At the moment, the matching strategy is determined by the user; we aim at introducing techniques to suggest a suitable strategy using a-priori analysis of the input ontologies [14], and make it adaptive during the matching process. Moreover, we are developing a clustered version of X-SOM, called *k*X-SOM, which exploits the intrinsic parallelism contained into the matching algorithms.

3.2 Comments on the OAEI 2007 procedure

The OAEI contest is well conceived, and has helped the improvement of the modules implemented in X-SOM. In our opinion, however, OAEI organizers should allow reparametrization of the tool, in order to better configure the prototype for each task.

3.3 Comments on the OAEI 2007 test cases

In our opinion, the benchmark track contains too many test cases in the systematic family (248-266). These tests reflect too "unreal" ontology design situations, while it will be more interesting to add test cases that include also complex alignments (i.e., mappings between complex definitions of concepts). In addition, it would be useful to introduce one or more test cases to analyze if different matching algorithms are able to avoid those mappings that produce contradictions in the ontological model.

3.4 Comments on the OAEI 2007 measures

Traditional precision and recall measures along with their combinations are the most suitable measures to evaluate the performance of ontology mapping tools. Moreover, it will be helpful to consider also the mappings among external resources.

3.5 Acknowledgment

We thank all the people and students that worked on X-SOM, in particular Alessandro Dalvit who executed the OAEI tests.

4 Conclusion

Ontology alignment and integration represents crucial aspects of the effort that computer science community is making to achieve systems interoperability. The OAEI contest represents a valuable opportunity to gather all the approaches and improve the current matching algorithms. Participating in the OAEI allowed the identification of the weaknesses of the X-SOM approach, in particular, problems have arisen with large ontologies and during the aggregation phase. It is our intention to address these weaknesses in order to improve our approach for OAEI 2008.

References

1. C. Bolchini, C. Curino, F. A. Schreiber, and L. Tanca. Context integration for mobile data tailoring. In *Proc. of the 7th Int. Conf. on Mobile Data Management (MDM'06)*. IEEE Computer Society Washington, DC, USA, 2006.
2. G. Orsi C. Curino and L. Tanca. X-som: Ontology mapping and inconsistency resolution. In *Proc. of the 4th European Semantic Web Conference (ESWC'07) - Poster Session*, Innsbruck, Austria, June 2007.
3. P. Cimiano and S. Staab. Learning by Googling. *ACM SIGKDD Explorations Newsletter*, 6(2):24–33, 2004.
4. W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. *Proc. of the 1st Int. Workshop on Information on the Web (IIWEB'03)*, pages 73–78, 2003.
5. C. Curino, G. Orsi, and L. Tanca. X-SOM: A Flexible Ontology Mapper. In *Proc. 1st Intl. Workshop on Semantic Web Architectures For Enterprises*, September 2007.
6. C. Curino, E. Quintarelli, and L. Tanca. Ontology-based information tailoring. In *Proc. 2nd IEEE Int. Workshop on Database Interoperability (InterDB 2006)*, pages 5–5, April 2006.
7. M. Ehrig, S. Staab, and Y. Sure. Bootstrapping ontology alignment methods with apfel. pages 1148–1149, 2005.
8. A. Gómez-Pérez, M. Fernández-López, and O. Corcho. *Ontological Engineering*. Springer, 2004.
9. W. Hu, N. Jian, Y. Qu, and Y. Wang. Gmo: A graph matching for ontologies. *K-Cap 2005 Workshop on Integrating Ontologies*, pages 43–50, 2005.
10. A. Kalyanpur, B. Parsia, E. Sirin, B. Cuenca-Grau, and J. Hendler. Swoop: A web ontology editing browser. *Journal of Web Semantics*, 2005.
11. C. Leacock, M. Chodorow, and G. A. Miller. Using corpus statistics and wordnet relations for sense identification. *Computational Linguistics*, 24(1):147–165, 1998.
12. J. Madhavan, P. A. Bernstein, P. Domingos, and A. Y. Halevy. Representing and reasoning about mappings between domain models. *Proc. of the 18th Nat. Conf. on AI*, 2002.
13. G. Miller. *WordNet*. MIT Press, 1998.
14. M. Mochol, A. Jentzsch, and J. Euzenat. Applying an analytic method for matching approach selection. In *Proc. of The 1st Workshop on Ontology Matching (OM2006)*.
15. H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner. Ontology-based integration of information: a survey of existing approaches. In *Proc. of the Int. Workshop on Ontologies and Information Sharing*, pages 108–117, 2001.

Appendix: Raw results

#	Name	Prec.	Rec.	Time (sec)
101	Reference alignment	1.00	0.98	59.206
102	Irrelevant ontology	NaN	NaN	48.392
103	Language generalization	1.00	0.98	60.100
104	Language restriction	0.97	1.00	62.408
201	No names	0.81	0.81	77.133
202	No names, no comments	0.82	0.82	69.696
203	No comments	1.00	0.97	65.871
204	Naming conventions	0.99	0.69	84.651
205	Synonyms	0.72	0.71	75.767
206	Translation	0.74	0.73	71.233
207		0.69	0.68	92.888
208		0.99	0.75	66.384
209		0.70	0.69	71.624
210		0.70	0.69	70.693
221	No specialisation	1.00	0.99	61.508
222	Flatenned hierarchy	1.00	0.98	74.516
223	Expanded hierarchy	1.00	0.98	91.145
224	No instance	1.00	0.98	58.956
225	No restrictions	1.00	0.98	45.680
228	No properties	1.00	1.00	17.235
230	Flattened classes	0.99	0.97	58.191
231	Expanded classes	1.00	0.97	59.549
232		1.00	0.99	54.451
233		1.00	1.00	16.366
236		1.00	1.00	15.356
237		1.00	0.98	57.156
238		1.00	0.99	67.714
239		0.97	1.00	16.436
240		0.97	1.00	22.687
241		1.00	1.00	14.682
246		0.97	1.00	14.521
247		0.97	1.00	21.739
248		0.75	0.75	71.264
249		0.60	0.60	67.246
250		0.18	0.18	20.118
251		0.45	0.45	72.949
252		0.49	0.49	98.557
253		0.54	0.54	68.091
254		0.03	0.03	19.909
257		0.12	0.12	19.319
258		0.32	0.32	78.671
259		0.33	0.33	100.424
260		0.03	0.03	19.963
261		0.03	0.03	34.414
262		0.00	0.00	19.336
265		0.00	0.00	18.999
266		0.03	0.03	32.471
301	Real: BibTeX/MIT	0.91	0.49	42.561
302	Real: BibTeX/UMBC	1.00	0.58	23.537
303	Real: Karlsruhe	0.96	0.73	70.359
304	Real: INRIA	0.96	0.87	48.097