

LogMap and LogMapLt Results for OAEI 2012

Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, and Ian Horrocks

Department of Computer Science, University of Oxford
{ernesto, berg, ian.horrocks}@cs.ox.ac.uk

Abstract. We present the results obtained by our ontology matching system LogMap and its ‘lightweight’ variant called LogMapLt within the OAEI 2012 campaign. The LogMap project started in January 2011 with the objective of developing a scalable and logic-based ontology matching system. This is our third participation in the OAEI and the experience has so far been very positive.

1 Presentation of the system

LogMap [10, 14] is a highly scalable ontology matching system with built-in reasoning and inconsistency repair capabilities. LogMap also supports (real-time) user interaction during the matching process, which is essential for use cases requiring very accurate mappings. To the best of our knowledge, LogMap is the only matching system that (1) can efficiently match semantically rich ontologies containing tens (and even hundreds) of thousands of classes, (2) incorporates sophisticated reasoning and repair techniques to minimise the number of logical inconsistencies, and (3) provides support for user intervention during the matching process.

LogMap is also available as a ‘lightweight’ variant called LogMapLt, which essentially skips all reasoning, repair and semantic indexation steps. Due to its simplicity, scalability and reasonable quality of its output, LogMapLt has been adopted as baseline in some OAEI tracks [19].

1.1 Technical challenges

Building a scalable, logic-based and interactive ontology matching presents important technical challenges. Moreover, these requirements are in some respects conflicting, and design choices require compromises between them. We next provide an overview of the technical challenges we have faced in the design of LogMap.

I. Computing Candidate Mappings. Computing mappings requires pairwise comparison of the entities in the vocabularies of the relevant ontologies (e.g., using a string matcher). This leads to a search space that is quadratic in the size of the ontologies (e.g., there are over 4 billion candidate mappings between FMA and NCI). For large ontologies, performing such huge number of pairwise comparisons is unfeasible in practice, even if we rely on the fastest available string matchers. Hence, reducing the search space of candidate mappings is a key challenge for a scalable ontology matching system.

II. Detection of unsatisfiable classes. Ontology $\mathcal{O}_1 \cup \mathcal{O}_2 \cup \mathcal{M}$ resulting from the integration of \mathcal{O}_1 and \mathcal{O}_2 via mappings \mathcal{M} may entail axioms that do not follow from

\mathcal{O}_1 , \mathcal{O}_2 , or \mathcal{M} alone. Many such entailments correspond to unsatisfiable classes, which are due to either erroneous mappings or to inherent disagreements between \mathcal{O}_1 and \mathcal{O}_2 . For example, the union of FMA, SNOMED and the UMLS [3] mappings between them (which are the result of careful manual curation) has over 6,000 unsatisfiable classes [13], and the number of unsatisfiable classes may be even higher when mappings are not subject to manual curation. Although state-of-the-art OWL 2 reasoners can efficiently classify existing large-scale biomedical ontologies individually (e.g., ELK [16] can classify SNOMED in a few seconds and HermiT [21] can classify FMA in less than a minute), the integration of these ontologies via mappings leads to challenging classification problems [9] (e.g., no reasoner known to us can classify the integration of SNOMED and NCI via mappings).

III. Repair of unsatisfiable classes. Standard justification-based repair techniques (e.g., [15, 23, 8]) can be used to repair the identified unsatisfiable classes in $\mathcal{O}_1 \cup \mathcal{O}_2 \cup \mathcal{M}$. These techniques have been implemented in mapping repair systems such as ContentMap [12] and Alcomo¹ [18]. The scalability problem, however, is exacerbated by the number of unsatisfiable classes to be repaired. For example, computing all justifications for just one out of the 6,000 unsatisfiable classes in the integration of FMA-SNOMED via UMLS mappings requires, on average, over 9 minutes using HermiT — even with the optimisation proposed in [24]; doing this for all unsatisfiable classes would require more than 6 weeks.

IV. Expert feedback during the matching process is important for use cases requiring very accurate mappings; however, smooth interaction with domain experts imposes very strict scalability requirements. Furthermore, feedback requests to a human expert should not be overwhelming and should be used only when strictly needed. Hence, it is crucial to reduce the number of feedback requests, on the one hand, as well as the delay between successive requests, on the other hand.

1.2 Technical approach

In order to meet these challenges, we have relied on the following key elements in the design of LogMap (see [10, 14] for details).

Lexical indexation. An inverted index is used to store the lexical information contained in the input ontologies. This index is the key to addressing *challenge I* since it allows for the efficient computation of an initial set of mappings of manageable size. Similar indexes have been successfully used in information retrieval and search engine technologies [2].

Logic-based module extraction. The practical feasibility of unsatisfiability detection and repair critically depends on the size of the input ontologies. To reduce the size of the problem, we exploit ontology modularisation techniques. Ontology modules with well-understood semantic properties can be efficiently computed and are typically much smaller than the input ontology [5, 17].

¹ Note that Alcomo also implements incomplete reasoning and repair techniques.

Propositional Horn reasoning. The relevant modules in the input ontologies together with (a subset of) the candidate mappings are encoded in LogMap using a Horn propositional representation. LogMap implements the classic Dowling-Gallier algorithm for propositional Horn satisfiability [6, 7], which can be exploited to detect unsatisfiable classes in linear time. Such encoding, although incomplete, allows LogMap to address *challenge II* soundly and efficiently.

Axiom tracking and greedy repair. LogMap extends Dowling-Gallier’s algorithm to track all mappings that may be involved in the unsatisfiability of a class. This extension is key to implementing a highly scalable greedy repair algorithm that can meet *challenge III*.

Semantic indexation. The Horn propositional representation of the ontology modules and the mappings are efficiently indexed using an interval labelling schema [1] — an optimised data structure for storing directed acyclic graphs (DAGs) that significantly reduces the cost of answering taxonomic queries [4, 22]. In particular, this semantic index allows us to answer many entailment queries over the input ontologies and the mappings computed thus far as an index lookup operation, and hence without the need for reasoning. The semantic index complements the use of a propositional encoding to address *challenges II-III* and it is the key to meeting *challenge IV*.

1.3 Adaptations made for the evaluation

LogMap’s algorithm described in [10, 14] has been extended with basic functionalities to support matching of instance data.

LogMap’s instance matching module is based on the same lexical indexation techniques used in LogMap to match classes. In order to discover additional instance mappings, LogMap also exploits the property assertions of the input ontologies to analyse the structure of their ABoxes.

In order to minimise the number of logical errors caused by the instance mappings, LogMap’s repair module is also used to detect and repair conflicts.

1.4 Link to the system and parameters file

LogMap² is open-source and released under GNU Lesser General Public License 3.0.³ Latest components and source code are available from the LogMap’s Google code page: <http://code.google.com/p/logmap-matcher/>.

LogMap distributions can be easily customized through a configuration file containing the matching parameters.

LogMap can also be used directly through an AJAX-based Web interface where matching tasks can be easily requested: <http://csu6325.cs.ox.ac.uk/>

² <http://www.cs.ox.ac.uk/isg/projects/LogMap/>

³ <http://www.gnu.org/licenses/>

Table 1: Results for Benchmark track.

System	biblio			bench1			bench2			bench2			finance		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
LogMap	0.73	0.45	0.56	1.00	0.47	0.64	0.95	0.49	0.65	0.99	0.46	0.63	0.95	0.47	0.63
LogMapLt	0.79	0.50	0.59	0.95	0.50	0.66	0.95	0.50	0.65	0.95	0.50	0.65	0.90	0.52	0.66

Table 2: Results for Anatomy track.

System	P	R	F	Time (s)
LogMap	0.92	0.845	0.881	20
LogMapLt	0.963	0.728	0.829	6

2 Results

In this section, we present the results obtained by LogMap and LogMapLt in the OAEI 2012 campaign.

2.1 Benchmark track

Ontologies in this track have been synthetically generated. The goal of this track is to evaluate the matching systems in scenarios where the input ontologies lack important information (e.g., classes contain no meaningful URIs or labels).

Table 1 summarises the average results obtained by LogMap and LogMapLt. Note that the computation of candidate mappings in LogMap and LogMapLt heavily relies on the similarities between the vocabularies of the input ontologies; hence, there is a direct negative impact in the cases where the labels are replaced by random strings.

2.2 Anatomy track

This track involves the matching of the Adult Mouse Anatomy ontology (2,744 classes) and a fragment of the NCI ontology describing human anatomy (3,304 classes). The reference alignment has been manually curated, and it contains a significant number of non-trivial mappings.

Table 2 summarises the results obtained by LogMap and LogMapLt. The evaluation was run on a machine with 4GB RAM and 2 cores.

2.3 Conference track

The Conference track uses a collection of 16 ontologies from the domain of academic conferences [25]. These ontologies have been created manually by different people and are of very small size (between 14 and 140 entities). The track uses two reference alignments RA1 and RA2. RA1 contains manually curated mappings between a subset of the 120 ontology pairs evaluated in the track. RA2 contains composed mappings, based on the alignments in RA1, between all the ontology pairs.

Table 3 summarises the average results obtained by LogMap and LogMapLt. The last column represents the total runtime on generating all 120 alignments. Tests were run on a laptop with Intel Core i5 2.67GHz and 4GB RAM.

Table 3: Results for Conference track.

System	RA1 reference			RA2 reference			Time (s)
	P	R	F	P	R	F	
LogMap	0.82	0.58	0.68	0.77	0.53	0.63	211
LogMapLt	0.73	0.5	0.59	0.68	0.45	0.54	44

Table 4: Results for Library track.

System	P	R	F	Time (s)
LogMap	0.688	0.644	0.665	95
LogMapLt	0.577	0.776	0.662	21

2.4 Multifarm track

This track is based on the translation of the OntoFarm collection of ontologies into 9 different languages [20]. Both LogMap and LogMapLt, as expected, obtained poor results since they do not implement specific multilingual techniques.

2.5 Library track

The library track involves the matching of the STW thesaurus (6,575 classes) and the TheSoz thesaurus (8,376 classes). Both of these thesauri provide vocabulary for economic and social sciences. Table 4 summarises the results obtained by LogMap and LogMapLt. The track was run on a machine with 7GB RAM and 2 cores.

2.6 Large BioMed track

This track aims at finding alignments between large and semantically rich biomedical ontologies such as FMA, SNOMED, and NCI [11]. UMLS Metathesaurus has been selected as the basis for the track reference alignments [3]. Since the UMLS mappings together with the input ontologies lead to numerous unsatisfiable classes, two refinements of the UMLS mappings have also been considered as reference alignments. These refinements have been generated using LogMap’s repair facility [10] and the Alcomodo debugging system [18]. The track has been split into nine tasks involving different fragments of FMA, SNOMED, and NCI.

LogMap has been evaluated with two configurations in this track. LogMap’s default algorithm computes an estimation of the overlapping between the input ontologies before the matching process, while the variant LogMap_{noe} has this feature deactivated.

Tables 5-7 summarises the results obtained by LogMap, LogMap_{noe} and LogMapLt. Precision and recall represent average values for the three reference alignments. The number of unsatisfiable classes as a consequence of reasoning (using HermiT [21]) with the input ontologies and the output mappings is also given.⁴ Note that LogMap, unlike LogMap_{noe}, failed to detect and repair a few unsatisfiable classes in the SNOMED-NCI matching problem since they were outside the computed ontology fragments. The track was run on a server with 16 CPUs and allocating 15GB RAM.

⁴ Since no OWL 2 reasoner can classify the integration of SNOMED and NCI via mappings [9], the Dowling-Gallier algorithm [6] for propositional Horn satisfiability was used instead.

Table 5: Results for the Large BioMed track: FMA-NCI tasks

Task 1: Small FMA and NCI fragments						
System	Size	Unsat.	P	R	F	Time (s)
LogMap	2,740	2	0.932	0.876	0.903	18
LogMap _{noe}	2,740	2	0.932	0.876	0.903	18
LogMapLt	2,483	2,104	0.945	0.806	0.870	8
Task 2: Big FMA and NCI fragments						
System	Size	Unsat.	P	R	F	Time (s)
LogMap	2,656	5	0.870	0.793	0.830	77
LogMap _{noe}	2,663	5	0.872	0.798	0.833	74
LogMapLt	3,219	12,682	0.729	0.806	0.766	29
Task 3: whole FMA and NCI ontologies						
System	Size	Unsat.	P	R	F	Time (s)
LogMap	2,652	9	0.860	0.783	0.819	131
LogMap _{noe}	2,646	9	0.866	0.787	0.825	206
LogMapLt	3,466	26,429	0.677	0.806	0.736	55

Table 6: Results for the Large BioMed track: FMA-SNOMED tasks

Task 4: Small FMA and SNOMED fragments						
System	Size	Unsat.	P	R	F	Time (s)
LogMap	6,164	2	0.910	0.667	0.769	65
LogMap _{noe}	6,363	0	0.910	0.688	0.784	63
LogMapLt	1,645	773	0.938	0.183	0.307	14
Task 5: Big FMA and SNOMED fragments						
System	Size	Unsat.	P	R	F	Time (s)
LogMap	6,292	0	0.833	0.623	0.712	484
LogMap _{noe}	6,450	0	0.837	0.642	0.727	521
LogMapLt	1,819	2994	0.848	0.183	0.302	96
Task 6: whole FMA and SNOMED ontologies						
System	Size	Unsat.	P	R	F	Time (s)
LogMap	6,312	10	0.828	0.621	0.710	612
LogMap _{noe}	6,406	10	0.816	0.621	0.706	791
LogMapLt	1,823	4938	0.846	0.183	0.301	171

Table 7: Results for the Large BioMed track: SNOMED-NCI tasks

Task 7: Small SNOMED and NCI fragments						
System	Size	Unsat.	P	R	F	Time (s)
LogMap	13,454	0*	0.897	0.649	0.753	221
LogMap _{noe}	13,525	0*	0.895	0.652	0.754	211
LogMapLt	10,947	61,269*	0.945	0.557	0.701	54
Task 8: Big SNOMED and NCI fragments						
System	Size	Unsat.	P	R	F	Time (s)
LogMap	12,142	3*	0.874	0.571	0.691	514
LogMap _{noe}	13,184	0*	0.879	0.624	0.730	575
LogMapLt	12,741	131,073*	0.812	0.557	0.661	104
Task 9: whole SNOMED and NCI ontologies						
System	Size	Unsat.	P	R	F	Time (s)
LogMap	13,011	16*	0.814	0.570	0.671	955
LogMap _{noe}	13,058	0*	0.811	0.570	0.670	1,505
LogMapLt	14,043	305,648*	0.737	0.557	0.634	178

Table 8: Results for Instance matching track.

System	Sandbox			IIMB		
	P	R	F	P	R	F
LogMap	0.94	0.94	0.94	0.94	0.91	0.93
LogMapLt	0.95	0.89	0.92	0.84	0.82	0.83

2.7 Instance matching

LogMap and LogMapLt have participated in the *Sandbox* and *IIMB* matching tasks. The Sandbox and IIMB datasets have been automatically generated by introducing a set of controlled transformations in an initial ABox, as a result Sandbox and IIMB contains 11 and 80 synthetic ABoxes, respectively.

Table 8 summarises the average results obtained by LogMap and LogMapLt. The results are quite promising considering that this is the first participation of LogMap in this track. Nevertheless, there is still room for improvement in order to deal with more challenging tasks.

3 General comments and conclusions

Comments on the results. LogMap’s main weakness is that the computation of candidate mappings relies on the similarities between the vocabularies of the input ontologies; hence, there is a direct negative impact in the cases where the ontologies are lexically disparate or do not provide enough lexical information.

Discussions on the way to improve the proposed system. LogMap is now a stable and mature system that has been made available to the community. There are, however, many exciting possibilities for future work. For example we aim at implementing multilingual features in order to be competitive in the Multifarm track. We also intend to extend LogMap’s instance matching module with more sophisticated techniques.

Comments on the OAEI 2012 measures. Although the *mapping coherence* is a measure already used in the OAEI we consider that is not given the required weight in the evaluation. Thus, developers focus on creating matching systems that maximize the F-measure but they disregard the impact of the generated output in terms of logical errors.

Acknowledgements. This work was supported by the Royal Society, the EPSRC project LogMap and the EU FP7 projects SEALS and Optique. We also thank the organisers of the OAEI evaluation campaigns for providing test data and infrastructure and Anton Morant and Yujiao Zhou who have also contributed to the LogMap project in the past.

References

1. Agrawal, R., Borgida, A., Jagadish, H.V.: Efficient management of transitive relationships in large data and knowledge bases. In: SIGMOD Rec. 18. pp. 253–262 (1989)

2. Baeza-Yates, R.A., Ribeiro-Neto, B.A.: *Modern Information Retrieval*. ACM Press / Addison-Wesley (1999)
3. Bodenreider, O.: The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research* 32, 267–270 (2004)
4. Christophides, V., Plexousakis, D., Scholl, M., Tourtounis, S.: On labeling schemes for the Semantic Web. In: *Int'l World Wide Web (WWW) Conf.* pp. 544–555 (2003)
5. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. *J. Artif. Intell. Res.* 31, 273–318 (2008)
6. Dowling, W.F., Gallier, J.H.: Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *J. Log. Prog.* 1(3), 267–284 (1984)
7. Gallo, G., Urbani, G.: Algorithms for testing the satisfiability of propositional formulae. *J. Log. Prog.* 7(1), 45–61 (1989)
8. Horridge, M., Parsia, B., Sattler, U.: Laconic and precise justifications in OWL. In: *Int'l Sem. Web Conf. (ISWC)*. pp. 323–338 (2008)
9. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I.: On the feasibility of using OWL 2 DL reasoners for ontology matching problems. In: *OWL Reasoner Evaluation Workshop (2012)*
10. Jimenez-Ruiz, E., Cuenca Grau, B.: LogMap: Logic-based and Scalable Ontology Matching. In: *Int'l Sem. Web Conf. (ISWC)*. pp. 273–288 (2011)
11. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I.: Exploiting the UMLS Metathesaurus in the Ontology Alignment Evaluation Initiative. In: *E-LKR Workshop (2012)*
12. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga, R.: Ontology integration using mappings: Towards getting the right logical consequences. In: *Eur. Sem. Web Conf. (ESWC)*. pp. 173–187 (2009)
13. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga, R.: Logic-based assessment of the compatibility of UMLS ontology sources. *J. Biomed. Sem.* 2 (2011)
14. Jiménez-Ruiz, E., Cuenca Grau, B., Zhou, Y., Horrocks, I.: Large-scale interactive ontology matching: Algorithms and implementation. In: *Eur. Conf. on Artif. Intell. (ECAI) (2012)*
15. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of OWL DL entailments. In: *Int'l Sem. Web Conf. (ISWC)*. pp. 267–280 (2007)
16. Kazakov, Y., Krötzsch, M., Simancik, F.: Concurrent classification of EL ontologies. In: *Int'l Sem. Web Conf. (ISWC)*. pp. 305–320 (2011)
17. Konev, B., Lutz, C., Walther, D., Wolter, F.: Semantic modularity and module extraction in description logics. In: *European Conf. on Artif. Intell. (ECAI)*. pp. 55–59 (2008)
18. Meilicke, C.: *Alignment Incoherence in Ontology Matching*. Ph.D. thesis, University of Mannheim (2011)
19. Meilicke, C., Svab-Zamazal, O., Trojahn, C., Jimenez-Ruiz, E., Aguirre, J., Stuckenschmidt, H., Cuenca Grau, B.: Evaluating ontology matching systems on large, multilingual and real-world test cases. In: *ArXiv e-prints (2012)*, <http://arxiv.org/abs/1208.3148v1>
20. Meilicke, C., Castro, R.G., Freitas, F., van Hage, W.R., Montiel-Ponsoda, E., de Azevedo, R.R., Stuckenschmidt, H., Šváb-Zamazal, O., Svátek, V., Tamin, A., Trojahn, C., Wang, S.: MultiFarm: a benchmark for multilingual ontology matching. *J. Web Sem.* (2012)
21. Motik, B., Shearer, R., Horrocks, I.: Hypertableau reasoning for description logics. *J. Artif. Intell. Res.* 36, 165–228 (2009)
22. Nebot, V., Berlanga, R.: Efficient retrieval of ontology fragments using an interval labeling scheme. *Inf. Sci.* 179(24), 4151–4173 (2009)
23. Schlobach, S., Huang, Z., Cornet, R., van Harmelen, F.: Debugging incoherent terminologies. *J. Autom. Reasoning* 39(3) (2007)
24. Suntisrivaraporn, B., Qi, G., Ji, Q., Haase, P.: A modularization-based approach to finding all justifications for OWL DL entailments. In: *Asian Sem. Web Conf. (ASWC) (2008)*
25. Šváb, O., Svátek, V., Berka, P., Rak, D., Tomášek, P.: OntoFarm: towards an experimental collection of parallel ontologies. In: *Int'l Sem. Web Conf. (ISWC). Poster Session (2005)*