

Holontology : results of the 2018 OAEI evaluation campaign

Philippe Roussille, Imen Megdiche, Olivier Teste, Cassia Trojahn

Institut de Recherche en Informatique de Toulouse, Toulouse, France
{firstname.name}@irit.fr

Abstract. This paper presents the results obtained by Holontology (Holistic ontology matcher) system in the OAEI 2018 campaign. We describe here its results in the tracks Anatomy, Conference, Taxon and Knowledge Graph. We report a general discussion on the results and on the future improvements of the system.

1 Presentation of the system

1.1 State, purpose, general statement

Holontology is a modular system based on LPHOM (Linear Program for Holistic Ontology Matching) [1]. While, as its predecessor LPHOM, the system remains a holistic ontology matching system (i.e., matching multiple ontologies simultaneously). Although the system has been designed to deal with holistic ontology matching, it is able as well to deal with pairwise ontology matching, as described here.

The system treats the ontology matching problem, at schema-level, as a combinatorial optimization problem. The problem is modeled through a linear program extending the maximum-weighted graph matching problem with linear constraints (matching cardinality, structural, and coherence constraints).

1.2 Specific techniques used

The way the tool works is detailed in the following steps:

1. The first step of Holontology is to load the ontologies and translate them into an internal structure, which can be annotated and edited, so that later treatment is done in a more efficient way. Each ontology is loaded independently, and its hierarchical structure is conserved in a format based on the three possible basic blocks of ontologies : classes, and properties (object and data). Each of its objects is represented as a Node, and, depending on its subtype, subclassed into a ClassNode, ObjectPropertyNode and DataPropertyNode. These Nodes are then enriched with AnnotationData.
2. The second step consists in a pre-processing step. Here, we expand camel case and title case into proper names. As the loading of the ontologies is done only once, we take advantage of Java ways of storing references, so we have no need to translate and cut matrices and we can use our hierarchies both for storage and computation.

3. The third step computes a cartesian product between all the entities of same type (classes and properties) in order to build a similar linear program using the same framework as LPHOM.
4. The fourth step leads a second wave of pre-processing, in such a way that new relations are created. For example, this is the step where we compute tense similarity (seeing whether or not an object property is passive or active towards its classes).
5. The fifth step applies a combination of different similarity measures (exact match, Levenstein, Jaccard, Lin), drawing both from the Annotated Data build in the previous steps, and the constraints based on ontologies themselves, onto the values drawn from the third step. We have tested the combination of similarity measures. For the results we report here, as expected, we obtain good values of precision with an exact match on the pre-processed annotations.
6. The sixth step converts the given constraints in a form that can be used by a solver. We choose CPLEX for its ability to perform optimizations, and we manage probable thresholds if needed. Unlike LPHOM, we provide all the constraints, and we don't apply cuts in this step.
7. In the seventh step, we take the given result from the solver and convert it in an alignment that can be exported in RDF.
8. In the (optional) eighth step, we combine the alignment result and run step five to seven if needed to reinforce the obtained results.

1.3 Adaptations made for the evaluation

Due to a huge incompatibility between some libraries used in the SEALS client and ours, we actually had to create a fully executable jar in the `conf/` directory, and call this from the SEALS bridge. The bridge grabbed the URI of the two ontologies passed as arguments, and wrote them to a file called "bridge-ontologies.url", then called the `holontology.jar` in the `conf` directory, waiting for its full execution. `holontology.jar` opens the created "ontologies.url", reads the two URIs, and does the processing. The result is written in the "result.alignment" file. The bridge then reads the result file and returned it to SEALS.

1.4 Link to the system and parameters file

Holontology, as its predecessor LPHOM, is not available as an open-source. This version is meant to be more modulable than Holontology, and handles the problem differently, by insisting on annotating data and using structures instead of matrices. The .zip for the SEALS jar can be retrieved at <https://cloud.irit.fr/index.php/s/gReZo8yaRDqdmjk>

1.5 Link to the set of provided alignments

The generated alignments are available at URL <https://cloud.irit.fr/index.php/s/hv3oALXN6fHuZWi>.

2 Results

The reader can refer to the OAEI web pages for the results of Holontology in the tasks Anatomy, Conference, Taxon and Knowledge Graph. Here, we provide a discussion and comments on our results.

2.1 Anatomy Track

Our results for the Anatomy track are summarized in Table 2.1. Compared to the evaluation of LPHOM in OAEI'2016¹, we observe that globally the quality of results decreases, for instance the F-measure loses 0.3 points. These results can be explained by two choices used in Holontology, first we only use exact match (we observe that Holontology returns only 456 alignments compared to LPHOM which returns 1555 alignments), second we don't cut results according to that. However, we observe that Holontology is 8 times faster than LPHOM.

Matcher	Runtime	Size	Precision	F-Measure	Recall	Recall+	Coherent
Holontology	265	456	0.976	0.451	0.294	0.005	-

Table 1. Results for Anatomy Track.

2.2 Conference Track

Our results for the Conference track are summarised in Table 2.2. Contrarily to the anatomy track, the results of Holontology are much more better than the results of LPHOM for the conference track. Globally, Holontology gains in recall for the different tasks in the track. We can explain that by the different pre-processing strategies that have been implemented in Holontology compared to LPHOM. By comparing the different tasks, the tool needs additional efforts to handle data and object properties that occur in the M2 tasks.

2.3 Complex Track (Taxon task)

Hontology is not able to deal with complex matching and has not been initially registered to this task. However, given that only 3 systems have been registered to the complex track, the organisers have also run the systems registered to Anatomy and Conference. Hence, the results reported for Hontology are in terms of simple generated alignments. Our results for Taxon task are summarised in Table 2.3. We have been obtained intermediate results, with a precision up to 0.22. However, for the set of given queries to be translated with the help of the generated alignments, our alignments were not useful, QWR (Query Well Rewritten) measure of 0.

¹ <http://oaei.ontologymatching.org/2016/results/anatomy/index.html>

Track	Rank	Prec.	F.5-measure	F1-measure	F2-measure	Recall
ra1-M1	8/13	0.88	0.78	0.67	0.59	0.54
ra1-M2	11/13	0.22	0.17	0.13	0.1	0.09
ra1-M3	10/13	0.78	0.69	0.59	0.52	0.48
ra2-M1	6/13	0.81	0.72	0.62	0.54	0.5
ra2-M2	11/13	0.07	0.05	0.03	0.02	0.09
ra2-M3	7/13	0.74	0.65	0.55	0.48	0.44
rar2-M1	9/13	0.8	0.72	0.63	0.56	0.52
rar2-M2	11/13	0.22	0.17	0.13	0.1	0.09
rar2-M3	9/13	0.73	0.65	0.56	0.49	0.45

Table 2. Results for conference

Time (s)	output corres.	eval. corres.	correct corres.	Global Prec.	Average Prec.	(1:1)	(1:n)	(m:n)	QWR
965	44	13	3	0.23	0.22	44	0	0	0.00

Table 3. Results for taxon

2.4 Knowledge Graph

Our results for the Knowledge Graph track are summarized in Table 2.4 and 2.4. For this track, Holontology proceeded faster than the other methods (including the baseline), but didn't include properties (probably, as the track organizers explained, because all properties are typed as `rdf:Property` and not subdivided into `owl:DatatypeProperty` and `owl:ObjectProperty`).

Time	#tracks	class				overall			
		Size	Prec.	F-m.	Rec.	Size	Prec.	F-m.	Rec.
318	9	16.8	0.80 (0.80)	0.83 (0.83)	0.87 (0.87)	18.8	0.80 (0.80)	0.17 (0.17)	0.10 (0.10)

Table 4. Global results for Knowledge-Graph

Track	Time	Size	Prec.	F-m.	Rec.
darkscape~oldschoolrunescape	23	42	0.75	0.22	0.13
runescape~darkscape	38	62	0.93	0.25	0.14
runescape~oldschoolrunescape	35	47	0.62	0.22	0.13
heykidscomics~dc	67	4	1.00	0.10	0.05
marvel~dc	26	4	1.00	0.21	0.12
marvel~heykidscomics	61	4	1.00	0.11	0.06
memory-alpha~memory-beta	26	2	0.00	0.00	0.00
memory-alpha~stexpanded	21	2	0.00	0.00	0.00
memory-beta~stexpanded	21	2	0.00	0.00	0.00

Table 5. Track results for Knowledge-Graph

3 General comments

Despite the fact that Hontology is an extended version of LPHOM that has participated in OAEI 2016, for its first participation, Hontology has intermediate results. Table 3

summarises the performance of Hontology in terms of ranking of the best systems in each task.

Track	Rank
Anatomy	14/14
Taxon	3/7
Knowledge graph	6/8 (overall)
Knowledge graph	1/8 (classes)
Conference	9.1/13 (average)

Table 6. Global rankings for Holontology per track

With respect to LPHOM, Hontology is a modular system that optimises the structures in memory. As for LPHOM, we model and express the matching problem through a set of constraints (cardinality, structural, and coherence constraints) applied on the results of a pre-processing and exact matching steps. We do not have applied any threshold on the generated alignments. As expected, using an exact match may improve precision in detriment of recall. With respect to these aspects, we plan to improve the criteria of selection of similarity measures and thresholds for our future participation, in particular with the aim of improving recall.

Other points include the fact that Hontology is a system designed to deal with holistic ontology matching at schema-level. Hence, it was not able to generated alignments for the tasks involving instance matching. We plan to implement instance matching strategies in future versions of the system. Furthermore, despite our optimisation over LPHOM, our system was not able to deal at all with the large ontologies in the Large-Bio and Phenotype tasks. We note, however, that we could deal with the Complex Taxon track in terms of volume because we do not treat the instances. We plan to address these points in the future.

With respect to the OAEI procedure, we focus on the tracks based on SEALS. However, as stated above, we have encountered problems for dealing with the incompatibilities of package versions in the SEALS dependencies. We have implemented a non-ideal solution and hope for the next evaluation this kind of issue will be fixed.

Finally, Hontology has been initially designed to deal with holistic ontology matching. However, there is no track in the campaign proposing the evaluation of such kind of matching approaches. In the future, it could be interesting to have a dedicated holistic track.

4 Conclusions

This paper has introduced the Hontology system and discussed the main points on the results of its first participation in OAEI campaigns. We have as well pointed out some directions for future improvements.

References

1. I. Megdiche, O. Teste, and C. Trojahn. LPHOM results for OAEI 2016. In *Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016.*, pages 190–195, 2016.