OWL-CTXMATCH in the OAEI 2006 alignment contest

Sławomir Niedbała

Institute of Computer Science, Silesian University of Technology, Gliwice, Poland
Slawomir.Niedbala@polsl.pl

Abstract. The OWL-CTXMATCH algorithm is one of the latest developments in the domain of ontology matching. It is next solution that represents a new family of algorithms performing so-called “semantic matching”. The paper presents the results of OWL-CTXMATCH obtained during its evaluation in the 2006 Campaign organized by Ontology Alignment Evaluation Initiative. The results have been preceded by a brief description of the algorithm and its implementation and completed with some conclusions.

1 Presentation of the system

1.1 State, purpose, general statement

The OWL-CTXMATCH algorithm has been designed to match OWL DL ontologies [8]. It is an OWL-specialized version of the CTXMATCH algorithm [2], which has been declared as a general algorithm designed to “discover semantic relationships across distinct and autonomous generic structures” [3]. The main requirement it imposes on the structures being matched is the necessity of labelling them with the natural language. The unique feature that both algorithms offer is that they perform so-called “semantic matching” [4] and as a result are able to recognize a broad range of relationships between matched entities, i.e. not only equivalence but also subsumption, disjointness and intersection.

This ability proved to be very useful in the domain of web services discovery [1], especially in the so-called “matchmaking” of web services by means of OWL-S Service Profiles [5]. There are number of algorithms in this domain, e.g. this implemented in OWL-S/UDDI Matchmaker [10], that require knowing exactly what kind of relationship holds between two or more given OWL entities. In this context particularly interesting are two kinds of relationships, i.e. equivalence and subsumption. However, the aforementioned matchmaking algorithms are not prepared for the problem of ontological heterogeneity. Their existing implementations simply ignore the fact that OWL-S Service Profiles might have been described by means of OWL classes and properties defined in different, unrelated ontologies. As a result, these implementations are not able to matchmake described in such a way web services, because they do not recognize hidden relationships that hold between unrelated OWL entities.

Recently there has been proposed a solution to this problem [7] that depends on the idea of adding in to existing tools a supplementary module that matches unrelated OWL entities on demand. The module is expected to be able to interpret ontologies mainly on the basis of their terminological parts, because there is no guarantee that these ontologies will provide sufficient set of instances. On the other hand it is expected that the entities of given ontologies will have been named with commonly used words, not with some unintelligible symbols or randomly generated texts. With this motivation in mind there has been developed the OWL-CTXMATCH algorithm, which has already been used in one of the latest OWL-S matchmakers called Cobra1Matchmaker [9].

1.2 Specific techniques used

OWL-CTXMATCH similarly as CTXMATCH realizes matching as a series of computations of relationships in which each computation is performed for every single pair of unfamiliar entities coming from both given structures. Each mapping is computed here in two steps. At first one there are built internal representations of both entities, by means of which the algorithm stores recognized interpretations. These interpretations are defined in the form of appropriate logical formulas. In OWL-CTXMATCH there have been proposed description logics formulas, which are more expressive than propositional logics formulas used in original CTXMATCH. The second step of finding single mapping amounts to computing what relationship holds between particular entities on the basis of their internal representations. Since OWL-CTXMATCH uses description logics formulas, this step in practice is realized by an external DL reasoner that initially merges both sets of formulas into one model, performs classification of it and finally determines what kind of relationship holds between counterparts for the particular entities.

As it has already been depicted in the paper [2] the essential impact on the way how in the CTXMATCH–based algorithms the given entities are interpreted has three levels of available knowledge. They are: lexical knowledge that defines meanings of the words used in the entities labels, domain knowledge that provides relations holding between senses associated with the given entities, and structural knowledge that in case of OWL-CTXMATCH is simply interpretation of the given OWL models. The current version of the algorithm uses solely WordNet dictionary [6] as a source of both lexical and domain knowledge and therefore is limited to the English language only. However, the algorithm can be easily modified in order to use other thesauri. Furthermore there is an intention to develop a new version of the algorithm that will be able to use other kinds of sources, e.g. general knowledge or reference ontologies like it has been described in the proposal of the CTXMATCH2 algorithm [11].

Since in the OWL structures there can be distinguished two primary elements, that is OWL classes (concepts) and OWL properties (roles), the OWL-CTXMATCH algorithm required designing two variants of matching procedures. In both cases the general idea of finding a mapping is the same, however matching of OWL properties is more complicated and, in effect, requires performing double quantity of computations while finding mappings between OWL classes is performed in an original one-step way. Another issue appeared during matching of OWL ontologies that needed to be resolved in OWL-CtxMatch was the problem of how to find mappings for classes and properties whose definitions are mutually dependent. For this reason the current version of the algorithm has simplified the issue by ignoring built-in `rdfs:domain` and `rdfs:range` properties. It means that using any of these restrictions will not affect the matching process at all. As a result definitions of OWL properties are never dependant on any OWL classes and therefore OWL-CtxMatch matches properties at first and classes afterwards, where it can use already found mappings between properties, on which given classes depend.

1.3 Adaptations made for the evaluation

Since the OWL-CtxMatch generates mappings in its internal form, these mappings have to be converted to the official alignment format for the purpose of OAEI 2006 Campaign. This task, however, gave rise to another problem, namely how to perform a proper conversion so the algorithm results would be well comparable with others. Assuming that all of the internal mappings are transformed to the alignment format, there would be produced very large resultant alignments during these conversions, because for each pair of entities coming from different ontologies there would exist one mapping tuple. Thus it would drastically deteriorate precision results achieved by the algorithm without getting any gain in recall values. For this reason in the current approach during the conversion there are transformed only those mappings, in which recognized relationship is equivalence. These filtered mappings are then written in the files named “OCM.rdf” (after the first letters of Owl-CtxMatch).

Apart from them there are also created supplementary files named “OCM_full.rdf”, in which there are additionally allowed mappings having subsumption relations. It is worth mentioning that although these files are extended with the “_full” suffix, they actually do not contain all the mappings. It probably would be more appropriate to name them “OCM_extended.rdf”, what might be reconsidered in the next versions of the application. Finally it must be emphasized that these supplementary files are not the part of submission and they serve for informative purposes only.

1.4 Link to the system and parameters file

The OWL-CtxMatch implementation along with all its required libraries and information how to run the application is available at the following URL:

http://www-zo.iinf.polsl.gliwice.pl/~niedbyk/oaei2006/

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

The alignments produced by the OWL-CtxMatch are available on the same web site as mentioned before, i.e. they can be found at:

http://www-zo.iinf.polsl.gliwice.pl/~niedbyk/oaei2006/

2 Results

2.1 Benchmark tests

The first part of the benchmark tests (#1xx) was quite easy for OWL-CtxMatch. The algorithm found all the required relationships, however apart from them it also discovered some additional mappings that caused the algorithm's precision has deteriorated slightly. These mapping appeared to be an effect of the conflicts between structural knowledge gained from OWL ontologies and domain knowledge provided by WordNet. For instance, the reference ontology defines an OWL class named `Book` as a subclass of `Reference`, while in the WordNet one of the senses associated with `reference` is a hyponym of (a kind of) one of the senses associated with the `book`. As a result OWL-CtxMatch deduces wrongly that both meanings are equivalent.

Other worth considering aspects of initial part of benchmark tests are their durations. Since the actual matching in OWL-CtxMatch is preceded with classifications of both input ontologies, the duration of all the computations is highly dependant on how long these classifications last. It is very noticeable that in case of matching the reference ontology with the wine ontology (test #102) the whole process took the algorithm as many as five hours. The reason was that the preliminary classification of the wine ontology was very time-consuming. In other cases matchings lasted considerably shorter, however they still required about six, seven minutes. Such long durations are connected with the fact that the algorithm computes all the mappings for each pair of entities coming from different ontologies. For instance it had to recognize more than three thousand of relationships in case of test #101 only.

The second part of the benchmark tests (#2xx) confirmed that the OWL-CtxMatch algorithm is strictly dependent on correct labelling ontology entities with the natural language words. It can be observed on the basis of the tests, in which recall results are equal or almost equal to zero. Such values have been obtained in tests #201, #202, #206, #207, #210 and also in #248-#266. In these tests labels of ontology entities have been either replaced by random texts or by their French translations (these non-English translations are not understandable by OWL-CtxMatch, which is a WordNet-based system).

Unexpectedly the algorithm achieved also poor recall results in the tests #205 and #209, in which synonyms have been used. It proves that WordNet, which currently fulfils a function of the source of lexical and domain knowledge, definitely needs to be aided with knowledge coming from other sources. It is also worth mentioning that in those two tests (#205 and #209) the matching durations have increased rapidly. It is fully understandable, because each synonym brings by means of its label new words and in case of polysemies their additional senses to internal formulas what causes OWL-CtxMatch to operate on the bigger internal representations.

The next alterations made in test ontologies, like removing instances (#224), modifying concepts hierarchy (#221-#223) or removing properties (#225, #228) etc. did not seem to affect OWL-CtxMatch matching abilities or affected them in a small extent. Furthermore, the good values of precision achieved by the algorithm in almost all tests of the second group confirmed the author in the conviction that OWL-CtxMatch can be useful not only in matchmaking of Web Services but also in other applications that require sophisticated ontology matching.

The last part of benchmark tests (#3xx) consisted of real ontologies and proved that the results achieved in previous two parts were not the outcome of special tuning of the algorithm to the their sets of synthetic ontologies. The obtained precisions of about 90% and the recalls of about 50% allow describing the OWL-CtxMatch algorithm as a solid and robust solution in the domain of ontology matching.

2.2 Directory and conference tests

The OWL-CtxMatch algorithm managed to perform all the directory tests without any serious problems. Since they are blind tests and there are not reference mappings available for this moment, it is difficult to comment obtained results. The similar situation of lacking reference mappings arises in reference to the conference tests, however in this case some of the ontologies have not been matched.

Firstly two out of ten conference ontologies (`confious.owl`, `OpenConf.owl`) turned out to be in OWL Full. It generally was impossible for OWL-CtxMatch to deal with them, since the algorithm is designed to matching OWL DL ontologies. Furthermore, the algorithm had unexpected problems with interpreting IASTED ontology (`iasted.owl`), even it was in OWL DL. The reason for this was the fact that the Pellet reasoner [12], which is used by the current implementation of OWL-CtxMatch, could not classify this ontology in a reasonable time. However, it must be said that there was not used the latest stable version of Pellet, but its nightly build from 2006-03-30/31. Since it is required for every ontology to be classified before actual matching begins, OWL-CtxMatch never could even have started to match anything with IASTED ontology. Pellet is also to blame for the fact that OWL-CtxMatch could not match `sigkdd.owl` with `cmt.owl`. This is the only case excluding IASTED ontology when the OWL-CtxMatch algorithm could not handle matching two proper OWL DL ontologies.

3 Conclusion

In the paper there has been presented the results of the OWL-CtxMatch algorithm obtained during its evaluation in the OAEI 2006 Campaign. These results allow stating that the examined algorithm is a reliable solution for matching OWL DL ontologies. However, the evaluation that has been carried out did not take into considerations all the aspects of the examined solution, in particular, its special ability to perform semantic matching. Since almost all of the provided reference mappings have only equivalence relationships, it was impossible to assess other kinds of mappings recognized by OWL-CtxMatch. Furthermore, the reference set of relationships was prepared based on the assumption that all the examined algorithms will compute one-to-one mappings, while OWL-CtxMatch generates a set of mapping tuples with the n:m multiplicity. These two essential problems require extending the current evaluation approach so it will be possible to correctly assess results of the algorithms that perform semantic matching. Obviously it will entail alterations in the proposed set of test ontologies as well.

Despite these shortcomings of the evaluation method, the algorithm still needs some further improvements. At first there is a need to refine the methods of obtaining lexical and domain knowledge. As it has already been mentioned, there is a successor of CTXMATCH that allows using reference ontologies as a source of domain knowledge [11]. Other aspect of improvements concerns the algorithm's efficiency. It is currently considered distributing computations of single mappings to other machines and introducing iterative method of matching procedure so it would allow fast pruning the pairs of entities that are definitely not correlated.

References

[1] Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., Orchard, D.: Web Services Architecture. (2004) http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/

[2] Bouquet, P., Serafini, L., Zanobini, S.: Semantic coordination: a new approach and an application. In Fensel, D., Sycara, K.P., Mylopoulos, J. (eds.): The Semantic Web - ISWC 2003. Proceedings. Lecture Notes in Computer Science, Vol. 2870. Springer, Berlin Heidelberg (2003) 130–145

[3] Bouquet, P., Serafini, L., Zanobini, S., Benerecetti, M.: An algorithm for semantic coordination. In Proceedings of the Semantic Integration Workshop at the Second International Semantic Web Conference. Sanibel Island, Florida, USA (2003) 21-26.

[4] Giunchiglia, F., Shvaiko, P.: Semantic Matching. The Knowledge Engineering Review, Vol. 18(3). (2003) 265-280

[5] Martin, D., Paolucci, M., McIlraith, S., Burstein, M., McDermott, D., McGuinness, D., Parsia, B., Payne, T., Sabou, M., Solanki, M., Srinivasan, N., Sycara, K.: Bringing Semantics to Web Services: The OWL-S Approach. In Proceedings of the 1st International Workshop on Semantic Web Services and Web Process Composition. San Diego, California, USA (2004) 26-42

[6] Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.:. Introduction to WordNet: An On line Lexical Database. International Journal of Lexicography, 3(4), (1990) 235 244.

[7] Niedbała, S.: Towards Semantic Matching of OWL-S Service Profiles. In Proceedings of International Workshop on Control and Information Technology. Ostrava, Czech Republic (2005)

[8] Niedbała, S.: OWL-CTXMATCH: Algorytm semantycznego dopasowywania ontologii OWL (OWL-CTXMATCH: Algorithm for Semantic Matching of OWL Ontologies; in Polish). In Proceedings of 2nd Conference on Bazy Danych: Aplikacje i Systemy. Ustroń, Poland (2006)

[9] Niedbała, S.: Odkrywanie semantycznych usług Web Service w środowisku nieskojarzonych ontologii OWL (Semantic Web Services Discovery in the Environment of Unfamiliar Ontologies; in Polish). In Proceedings of 13th Conference on Sieci Komputerowe. Zakopane, Poland (2006)

[10] Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.P.: Semantic Matching of Web Services Capabilities. In Proceedings of the 1st International Semantic Web Conference. Sardinia, Italy (2002) 333-347

[11] Sceffer, S., Serafini, L., Zanobini, S.: Semantic Coordination of Hierarchical Classifications with Attributes. Technical Report no 706, Department of Information and Communication Technology, University of Trento, Italy (2004).
[12] Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A Practical OWL-DL Reasoner. Technical Report 2005-68. University of Maryland, Institute for Advanced Computer Studies, (2005)

Appendix: Raw results

Matrix of results

The following results have been achieved by the OWL-CtxMatch Java 5 implementation that was using the following external libraries: Pellet OWL reasoner (nightly build from 2006-03-30/31) and all its dependants, Jena 2.3 and all its dependants (included in Pellet package), Java WordNet Library 1.3 and its dependant. Thanks to the JWNL library, OWL-CtxMatch was able to communicate with WordNet 2.0, which is also required by the application. The presented below time values were obtained on a dual-processor machine (2 x Intel Xeon 3.06 GHz) with 1 GB RAM and Windows 2003 Server installed on it. The OWL-CtxMatch application was executed in the "Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_08-b03)" with 256 MB of initial heap space and 768 MB of maximum heap space. However, the obtained time values are somewhat approximate, since the reference ontology was initially classified just once (not in every test) and, moreover, the host server was not dedicated to performing ontology matching tasks only. The presented values of precision and recall have been computed by designed for this purpose the "Ontology alignment API and implementation" tool (version 2.4+ from 10-07-2006) recommended by OAEI.

#	Name	Prec.	Rec.	Time
101	Reference alignment	0.96	1.00	00:07:32
102	Irrelevat ontology	NaN	NaN	05:06:25
103	Language generalization	0.92	1.00	00:06:00
104	Language restriction	0.98	1.00	00:06:02
201	No names	1.00	0.01	00:02:59
202	No names, no comments	1.00	0.01	00:02:52
203	No comments	0.97	1.00	00:05:11
204	Naming conventions	0.96	0.90	00:05:14
205	Synonyms	0.85	0.23	00:17:54
206	Translation	1.00	0.23	00:07:17
207		1.00	0.23	00:06:57
208		0.95	0.90	00:05:15
209		0.85	0.23	00:20:59
210		1.00	0.23	00:07:13
221	No specialisation	0.91	0.98	00:04:38
222	Flatenned hierachy	0.93	1.00	00:04:53
223	Expanded hierarchy	0.97	0.96	00:08:59

224	No instance	0.96	1.00	00:04:59
225	No restrictions	0.93	1.00	00:04:04
228	No properties	0.75	1.00	00:01:05
230	Flatenned classes	0.95	1.00	00:04:58
231		0.97	1.00	00:05:06
232		0.99	0.98	00:04:30
233		0.94	0.97	00:01:01
236		0.87	1.00	00:01:04
237		0.98	1.00	00:04:36
238		0.93	0.97	00:08:41
239		0.74	1.00	00:01:07
240		0.85	0.88	00:02:52
241		0.86	0.97	00:00:59
246		0.85	1.00	00:01:08
247		0.81	0.88	00:02:51
248		1.00	0.01	00:02:45
249		1.00	0.01	00:02:46
250		NaN	0.00	00:00:52
251		1.00	0.01	00:03:22
252		1.00	0.01	00:11:17
253		1.00	0.01	00:03:10
254		NaN	0.00	00:01:03
257		NaN	0.00	00:01:04
258		1.00	0.01	00:03:22
259		1.00	0.01	00:10:36
260		0.00	0.00	00:01:11
261		0.00	0.00	00:02:47
262		NaN	0.00	00:01:01
265		0.00	0.00	00:01:12
266		0.00	0.00	00:02:48
301	BibTeX/MIT	0.91	0.52	00:02:29
302	BibTeX/UMBC	0.95	0.40	00:02:16
303	Karlsruhe	0.77	0.41	00:10:59
304	INRIA	0.92	0.63	00:07:03