ASE Results for OAEI 2012

Konstantinos Kotis[1], Artem Katasonov[1], Jarkko Leino[1]

[1] VTT Technical Research Centre of Finland, Tampere, FI
{Ext-konstantinos.kotis, Artem.Katasonov, Jarkko.Leino}@vtt.fi

Abstract. This paper presents ASE (Aligning Smart Entities) tool for the automated alignment of OWL domain ontology definitions in the context of Internet of Things (IoT). The effort is based on experience gained by the development of AUTOMSv2 for OAEI 2012. The development process of this tool has been driven by our motivation to use the ontology alignment functionality as part of the Smart Proxy approach for the matchmaking of IoT entities. More specifically, ASE supports the automated deployment of applications on environments that IoT devices (sensors and actuators) have been already deployed. This paper presents the alignment approach followed towards developing the tool and the official results obtained for OAEI 2012 campaign.

1 Presentation of the system

1.1 State, purpose, general statement

ASE (Aligning Smart Entities) is an automated ontology alignment tool based on AUTOMSv2 tool (http://ai-lab-webserver.aegean.gr/kotis/AUTOMSv2), a baseline tool we have developed for OAEI 2012 campaign. It computes 1:1 (one to one) alignments of two input domain ontologies in OWL, discovering equivalence and subsumption axioms between ontology elements, both classes and properties. The features that this tool integrates are summarized in the following points:

- It is implemented with the widely used open source Java Alignment API [1]
- It synthesizes lexical and lexicon-based alignment methods, using union aggregation operator
- It integrates state-of-the-art alignment methods with standard and extended methods from the Java Alignment API
- Implements a language translation method for non-English ontology elements

Comparing with AUTOMSv2, in ASE

This work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme. This Programme is supported by the Marie Curie Co-funding of Regional, National and International Programmes (COFUND) of the European Commission

a) We do not implement a profiling and configuration strategy, but instead we use a fixed synthesis method based on experience and observation of AUTOMSv2 behavior and also on specific performance requirements that the application domain of IoT and the specific Smart Proxy approach have been implied,
b) We implement the discovery of subsumption relations between concept/property pairs, in addition to equivalences,
c) We implement a new method for translating Non-English ontologies, a method that is based on the Microsoft Bing Translator API
d) We implement some utility functions for handling compound terms

The problem of computing alignments between ontologies can be formally described as follows: Given two ontologies $O_1 = (S_1, A_1)$, $O_2 = (S_2, A_2)$ (where S_i denotes the signature and A_i the set of axioms that specify the intended meaning of terms in S_i) and an element (class or property) E_i^1 in the signature S_1 of O_1, locate a corresponding element E_j^2 in S_2, such that a mapping relation (E_i^1, E_j^2, r) holds between them. *r* can be any relation such as the equivalence (≡) or the subsumption (⊑) axiom or any other semantic relation e.g. meronym. For any such correspondence a mapping method may relate a value γ that represents the preference to relating E_i^1 with E_j^2 via *r*. If there is not such a preference, we assume that the method equally prefers any such assessed relation for the element E_1. The correspondence is denoted by $(E_i^1, E_j^2, r, \gamma)$. The set of computed mapping relations produces the mapping function $f{:}S_1 \rightarrow S_2$ that must preserve the semantics of representation: i.e. all models of axioms A_2 must be models of the translated A_1 axioms: i.e. $A_2 \models f(A_1)$.

ASE can be seen as a subversion of AUTOMSv2 ontology alignment tool, in the sense that it uses a specific synthesis configuration of AUTOMSv2 alignment methods. The synthesis of alignment methods that exploit different types of information may discover different types of relations between elements have been already proved to be of great benefit [2, 5]. ASE configuration is based on the requirement that the related input ontology definitions in the application domain that this tool is used are very often flat (no structure), have no instances (unpopulated), have very few concepts/properties (1 to 5 in most cases), have no expressive axioms and compound terms are very common.

In ASE we follow a modern synthesis strategy, which performs composition of results at different levels: the resulted alignments of individual methods are combined using specific operators, e.g. by taking the union of results. Given a set of *k* alignment methods (e.g. string-based, WordNet-based), each method computes different confidence values concerning any assessed relation (E_1, E_2, r). The synthesis of these *k* methods aims to compute an alignment of the input ontologies, with respect to the confidence values of the individual methods. Trimming of the resulted correspondences in terms of a threshold confidence value is also performed for optimization.

The alignment algorithm followed in this work is outlined in the following steps:

- Step 0: If non-English names of labels of entities are detected, translate input ontology into an English-language copy of it.

– Step 1: For each integrated alignment method k compute correspondence (E_i^1, E_j^2, r, γ) between elements of the two domain ontologies.
– Step 3: Apply trimming process by allowing agents to change a variable threshold value (of γ) for each alignments set S_k or for the alignments of a synthesized method
– Step 4: Apply synthesis of methods at different levels (currently using union aggregation operator) to the resulted set of alignments S_k .

The proposed ontology alignment approach considers most of the challenges in ontology alignment research [3, 5]. Consider two alignment methods (Figure 1), *m* and *m'*, also called matchers, that are selected based on a fixed synthesis configuration method and used for aligning two input ontologies *o* and *o´*. In case of translation needed, this is performed before entering *m* and *m´* respectively. The resulting alignments are aggregated/merged in *a*, using an aggregation operator (union is the current one used), resulting in another alignment *A´´´* which will be improved by another alignment method *m''* resulting to the final alignment *A´´´´*.

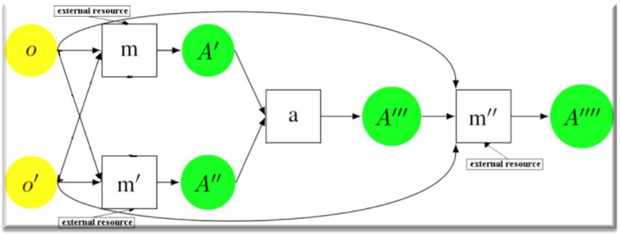

Fig. 1. General description of the ontology alignment process [5]

1.2 Specific techniques used

The tool has been developed by re-using AUTOMSv2 and Alignment API methods and libraries. Specifically, ASE synthesis configuration method merges the alignments of four synthesized alignment methods as described in the following paragraphs, having the first two dedicated to the computation of equivalences and the last two for the computation of subsumptions between ontology entities.

1. *Level 1 (for equivalences)*: Synthesis of three string-based similarity methods, one for each type of entity information i.e. names, labels and comments. For names similarity we use "smoaDistance" from Alignment API, for labels and comments similarity we use COCLU-based methods from AUTOMSv2. For each method a different threshold value is set (0.987 for COCLU-based and 0.82 for SMOA).
2. *Level 2 (for equivalences)*: Synthesis of two WordNet-based similarity methods for discovering synonyms between concept/property pairs, one for each type of entity information i.e. names and labels. For names similarity we use “basicSynonymySimilarity” from Alignment API and for labels we use our own method that is however based on the same basic synonym similarity approach.

3. *Level 3 (for subsumptions)*: Synthesis of two WordNet-based similarity methods for discovering subsumption relations between concept/properties, one for each direction i.e. a>b and a<b. We have developed these custom in-house methods only for labels, and totally depended on WordNet. So, if a hyperonym or hyponym relation between two terms exist in WordNet lexicon, then we conclude a subsumption axiom between the related ontology classes/properties.
4. *Level 4 (for subsumptions)*: Synthesis of two string-based similarity methods for discovering subsumption relations between concept/properties, one for each direction i.e. a>b and a<b. We have developed these custom in-house methods only for labels, and totally depended on the heuristic of compound terms such as: if there is a compound term (e.g. shortName) such as the right-most part of it can be matched to a non-compound term (e.g. name), then we can introduce a subsumption relation between these two such as the compound term is more specific than the non-compound e.g. shortName < Name (i.e. a short name is a kind of name).

The String Matching for Ontology Alignment (SMOA) method utilizes a specialized string metric "smoaDistance" for ontology alignment, first published in ISWC 2005 conference [6].

The WordNet-based string-based similarity distance 'basicSynonymySimilarity' computes the similarity of two terms based in their synonymic similarity, i.e. if they are synonyms in WordNet lexicon (returns '1' if term-2 is a synonym of term-1, else returns a BasicStringDistance similarity score between term-1 and term-2).

The state-of-the-art string similarity distance method COCLU, initially integrated in AUTOMS [4] and in other implementations using the AUTOMS-F API [7] is a partition-based clustering algorithm which divides data into clusters and searches the space of possible clusters using a greedy heuristic.ASE completely re-implements it and uses it in two different modes, i.e. in labels-mode and in comment-mode.

The large dependency of our alignment methods in an external resource such as WordNet is due to the specific requirement of the application domain that ASE is used in i.e. ontologies are very often flat (no structure), have no instances (unpopulated), have very few concepts/properties (1 to 5 in most cases), have no expressive axioms and compound terms are very common.

1.3 Link to the system and to the set of provided alignments (in align format)

ASE web page (short description, the system and OAEI results) is currently hosted at http://ai-lab-webserver.aegean.gr/kotis/ASE.

2 Results

The results reported in OAEI 2012 contest has been computed with an ASE version that does not integrate the methods for discovering subsumption relations between entities. This was decided due to the nature of the 'refaligns' provided by some organizers for some datasets. For instance, in Benchmark track, although a meaningful alignment between shortName and Name should have been included in the reference alignments with a subsumption relation (a ShortName is a Name), this was not the case. So, in order to avoid low precision due to this matter, we decided to exclude the capability of computing subsumption alignments for all tests.

2.1 Benchmark 2012

The Benchmark results for OAEI 2012 (http://oaei.ontologymatching.org/2012/benchmarks/index.html) indicated that ASE could not perform high in terms of precision (ranging between 0.27 and 0.72) but stay at the same levels as our AUTOMSv2 in terms of recall (ranging between 0.51 and 0.54) for the four out of five domains (see Table 1). For the last domain, i.e. finance (blind test), the tool did not compute any results. The low precision results however were related to additional mappings that have been recorded in the output alignment string, computed by one third-party method we reused (smoaDistance in Alignment API) which also aligns instances that are found in the ontologies (aligned entities can be classes, properties, and instances). At the same time, the reference alignments of Benchmark do not contain mappings of instances.

Having said that, since it is based in AUTOMSv2 alignment methods and Alignment API framework, we can expect that the corrected version will approximate at least the precision scores of AUTOMSv2 for this track (since AUTOMSv2 is the baseline for ASE development). This issue can be also supported by the fact that ASE computes the higher precision (0.72) for those datasets that have no (or the less) instances of all datasets i.e. benchmark-2.

Table 1. Scores for Benchmark track 2012

	Precision	F-measure	Recall	Runtime(s):
biblio	0.49	0.51	0.54	26
benchmark-2	0.72	0.61	0.53	69
benchmark-3	0.27	0.36	0.54	690
benchmark-4	0.4	0.45	0.51	276
finance	n/a	n/a	n/a	n/a

2.2 Conference 2012

The Conference results for OAEI 2012 (http://oaei.ontologymatching.org/2012/conference/index.html) indicated that ASE could perform higher in terms of precision (range between 0.61 and 0.63) and lower for recall (range between 0.4 and 0.43).

ASE failed to generate 6 alignments out of 120 testcases. Improved version delivered after deadline succeeded to generate all alignments (with improved scores, as in AUTOMSv2) however because it was delivered after deadline (and precision and recall performance was different) official results are reported according to initial submitted version. Runtime is reported according to the latest version which does not differ with the initial version much.

In this paper we decided to present (see Table 2), only the results generated with the official version of our tool (before the deadline of the contest), and not the one generated with an improved version (fixing unexpected third-party library crash) submitted after the deadline. This decision was made due to the feedback that we received from organizers of this track.

Table 2. Scores for Conference track 2012

Official (before deadline)				
	Precision	F-measure	Recall	Runtime(ms)
r1	0.63	0.51	0.43	104371
r2	0.61	0.48	0.4	104371

Comparing to AUTOMSv2 results for this track, ASE has generally an improved performance (f-measure is higher for both subtests), based mainly on the higher recall scores that we obtained. Also, runtime is quite improved (almost ¼ of AUTOMSv2 runtime).

Finally, we argue that if ASE was running on its full version, i.e. integrating also the methods for discovering subsumption relations between entities, it would have been achieved higher scores (sacrificing however performance in terms of runtime).

2.3 MultiFarm 2012

ASE was not able to compute official Multifarm results for OAEI 2012 (http://www.irit.fr/OAEI/). That was due to an unexpected crash of our third-party on-line translation API (Bing Translator) at the time of ASE execution by organizers.

Although we have immediately replaced this library with the one we use in AUTOMSv2, produced results for OAEI 2011.5 and OAEI 2012 campaigns, and obtained results also with ASE for this dataset, we do not report them here. In this paper we decided to present results generated with the official version of our tool (before the deadline of the contest) and not the ones generated with an improved version (fixing unexpected third-party library crash) submitted after the deadline. That decision was made due to the feedback and recommendation that we received from organizers of this track.

Table 3. Scores for MultiFarm track 2012

Unofficial results (after deadline)			
	Runtime	**Precision**	**Recall**
Lower	2687	0.15	0.00
Higher	237971	0.93	0.57
Average	18570	0.63	0.31

Having said that, from the results we obtained with the fixed unofficial version, we were able to gather good results (ranging between 0.15 and 0.93 for precision, 0 and 0.57 for recall, with largest runtime 237971s, and averages for precision=0.63, recall=0.31 and runtime=18570s), results that could be easily compared to AUTOMSv2 results for this track.

3 Comments

As already stated, the aim of this development experience, as with our baseline tool AUTOMSv2, was not to develop a tool to compete with others in terms of precision and recall. Instead, we aimed at the development of a subversion of AUTOMSv2 in order to fit in our application domain of IoT. Nevertheless, ASE obtained some good results (although not with the official OAEI 2012 version). As a general comment, ASE sacrificed precision (not much of recall though) for speed, since it uses only a subset of the alignment methods implemented in AUTOMSv2.

The following table summarizes the features of ASE tool:

Num. of input ontologies:	2
Ontology Elements:	Classes, Properties, Instances
Mapping cardinality:	1:1
Formal Language:	OWL
Relation:	=, <, >
Confidence:	[0, 1]
Natural Language:	EN, DE, FR, NL, ES, PT

ASE results could have been better (if using the latest unofficial version that we submitted after the deadline) and computation of results could have been performed also for other tracks (Library, Anatomy, LargeBio). We experienced a lot of unexpected difficulties with bugs appeared last minute in third-party libraries such as in Alignment API, COCLU string similarity method, WebTranslator API, and Microsoft Bing Translator API.

ASE is participating in this contest with its first prototype version. We plan to optimize its performance by testing and adapting new configurations of synthesized methods in a more efficient manner, always having AUTOMSv2 as our baseline tool.

In our future plans it is also the creation of a custom dataset and reference alignments using ontologies for the specific domain of IoT and Smart Environments. This is needed in order to better explore the requirements of such domain-specific evaluation of an ontology alignment tool. As it has been already stated, ASE must be evaluated in its context i.e. using ontologies that are very often flat (no structure), have no instances (unpopulated), have very few concepts/properties (1 to 5 in most cases), have no expressive axioms and compound terms are very common.

4 Conclusion

This paper presented ASE tool and official evaluation results obtained for OAEI 2012 contest. The effort was based on experience gained by the development of

AUTOMSv2 for OAEI 2011.5 and OAEI 2012. The development process of this tool was driven by our motivation to use the ontology alignment functionality as part of the Smart Proxy approach for the matchmaking of Internet of Things entities. In this paper we decided to present results generated with the official version of our tool (before the deadline of the contest) and not the ones (better in some cases) generated with the improved version (fixing unexpected third-party library crashes) submitted after the deadline. That decision was made due to the feedback and recommendation that we received from organizers of this track.

Acknowledgements

We thank all organizers for the valuable feedback and assistance towards delivering the presented results. We also acknowledge the work of developers/researchers in AUTOMS, AUTOMS-F and SMOA.

References

1. David, J., Euzenat, J., Scharffe, F., Trojahn dos Santos, C.: The Alignment API 4.0, Semantic Web - Interoperability, Usability, Applicability, 2(1):3-10, IOS Press (2011)
2. Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., Trojahn, C.: Ontology Alignment Evaluation Initiative: six years of experience, J. Data Semantics 15: 158-192 (2011)
3. Kotis, K., Lanzenberger, M.: Ontology Matching: Current Status, Dilemmas and Future Challenges. In: International Conference of Complex, Intelligent and Software Intensive Systems, pp. 924-927 (2008)
4. Kotis, K., Valarakos, A., Vouros, G. A.: AUTOMS: Automating Ontology Mapping through Synthesis of Methods, In: International Semantic Web Conference, Ontology Matching International Workshop, Atlanta USA (2006)
5. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges, IEEE Transactions on Knowledge and Data Engineering, 08 Dec. 2011. IEEE computer Society Digital Library. IEEE Computer Society, http://doi.ieeecomputersociety.org/10.1109/TKDE.2011.253
6. Stoilos, G., Stamou, G., Kollias, S.: A String Metric for Ontology Alignment. In: International Semantic Web Conference (2005)
7. Valarakos, A., Spiliopoulos, V., Kotis K., Vouros, G. A.: AUTOMS-F: A Java Framework for Synthesizing Ontology Mapping Methods, In: International Conference i-Know, Graz, Austria (2007)