AUTOMSv2 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 AUTOMSv2 effort towards building a tool for the automated alignment of domain ontologies. The developed tool is a result of our motivation to rebuild AUTOMS tool (presented in OAEI 2006) by putting together a) a well-known, widely used and continuously evolving/maintained alignment framework b) the synthesis of state-of-the-art alignment methods, c) a modern approach of synthesizing methods using profiling and configuration strategies, and d) multilingual support. The aim of this experience was not to compete with other tools in precision and recall but to re-develop AUTOMS using the abovementioned technologies and methods. Nevertheless, AUTOMSv2 obtained satisfactory results when compared with tools of OAEI 2011 and 2011.5 campaigns.

1 Presentation of the system

1.1 State, purpose, general statement

AUTOMSv2 is an automated ontology alignment tool based on its early version (AUTOMS) in 2006 [4]. It computes 1:1 (one to one) alignments of two input domain ontologies in OWL, discovering equivalences between ontology elements, both classes and properties. The features that this new version integrates are summarized in the following points:

- It is implemented with the widely used open source Java Alignment API [1]
- It synthesizes alignment methods at various levels and types (lexical, structural, instance-based, vector-based, lexicon-based) with the capability to aggregate their alignments using different aggregation operators (union, Pythagorean means)
- It implements an alignment-methods' configuration strategy based on ontology profiling information (size, features, etc.)
- It integrates state-of-the-art alignment methods with standard Alignment API methods
- Implements a language translation method for non-English ontology elements

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

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 ($\equiv$) or the subsumption ($\sqsubseteq$) 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. $A_2 \models f(A_1)$.

The synthesis of alignment methods that exploit different types of information (lexical, structural, and semantic) and may discover different types of relations between elements has been already proved to be of great benefit [2, 5]. Based on the analysis of the characteristics of the input ontology definitions, i.e. the profiling of ontologies, our approach provides different configurations (syntheses) of alignment methods. The analysis of input ontologies is based on their size, the existence of individuals or not, the existence of class/properties annotations e.g. labels, and the existence of entity names with an entry in WordNet lexicon. Part of the profiling is also a translation method that supports the translation of classes/properties annotations if these are given in a non-English language.

In the presented work we follow a modern synthesis strategy, which performs composition of results at different levels (see Figure 1): the resulted alignments of individual methods are combined using specific operators, e.g. by taking the union or intersection of results, intersection of results or by combining the methods' different confidence values with weighing schemas. Given a set of k alignment methods (e.g. string-based, vector-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 our work is outlined in the following steps:

- Step 1: Analyze ontology definitions to be aligned (profiling step) and assign the correspondent configuration of alignment methods to be used (configuration step). If needed, translate ontology into an English-language copy of it.
- Step 2: For each integrated alignment method k compute correspondence $(E_i^1, E_j^2, r, \gamma)$ between elements of the two domain ontologies.
- Step 3: Apply trimming process by allowing agents to change a variable threshold value 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] but emphasizes the alignment methods selection and synthesis.

1.2 Specific techniques used

The tool has been developed from scratch, reusing some of the alignment methods already provided within the Alignment API. Other state-of-the-art methods such as the COCLU string-based and the LSA vector-based methods implemented in AUTOMS [4] using the AUTOMS-F API [7] have been re-implemented using the new API. The instance-based and structure-based alignment methods have been also implemented from scratch. The detailed description of the alignment methods have been presented already in previously published works [4, 6, 7]. The integrated string-based methods are used in two different synthesized methods and in one single method. All three methods use class and property names as input to their similarity distance metrics.

The first synthesized method, synthesizes the alignments of two string-based similarity distance methods distributed with the Alignment API, namely, the 'smoaDistance' method and the 'levenshteinDistance'. A general Levenshtein distance between two strings is defined as the minimum number of edits needed to transform one string into the other, with the allowable edit operations being insertion, deletion, or substitution of a single character. The one re-used from the Alignment API is a version of the general distance metric, based on the Needleman Wunsch distance method. The String Matching for Ontology Alignment (SMOA) method utilizes a specialized string metric for ontology alignment, first published in ISWC 2005 conference [6].

The second synthesized method, synthesizes the alignments of two WordNet-based string-based similarity distance methods of the Alignment API, namely, the 'basicSynonymySimilarity' and the 'cosynonymySimilarity'. The first 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), and the second computes the proportion of common *synsets* between them, i.e. the proportion of common synonyms shared by both terms.

The third one is a single method that is implemented based on 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]. Since AUTOMSv2 completely re-implements it, it is used in two different modes, i.e. in names-mode and in labels-mode, according to the type of input ontologies that the profiling method will return. COCLU is a partition-based clustering algorithm which divides data into clusters and searches the space of possible clusters using a greedy heuristic.

Regarding vector-based alignment methods, AUTOMSv2 integrates two LSA-based methods, versions of the original HCONE-merge alignment method implemented in AUTOMS [4]. The first version is based on LSA (Latent Semantic Analysis) and WordNet and the second just in LSA. In the first one, given two ontologies, the algorithm computes a morphism between each of these two ontologies and a “hidden intermediate” ontology. This morphism is computed by the Latent Semantic Indexing (LSI) technique and associates ontology concepts with WordNet senses. Latent Semantic Indexing (LSI) is a vector space technique originally proposed for information retrieval and indexing. It assumes that there is an underlying latent semantic space that it estimates by means of statistical techniques using an association matrix (n×m) of term-document data (WordNet senses in this case). The second version of this method is based on the same idea but instead of exploiting WordNet senses it builds the term-document matrix from the concepts' names/labels/comments and their vicinity (properties, direct super-concepts, direct subconcepts) of the input ontologies. The similarity between two vectors (each corresponding to class name and annotation as well as to its vicinity) is computed by means of the cosine similarity measure.

Finally, two more methods, a structure-based and an instance-based method, are integrated, based on the general principle that two classes can be considered similar if a percentage of their properties or their instances has been already considered to be similar. The similarity of properties and instances is computed using a simple string-matching method (Levenshtein). In case structure and instances are not common in the input ontologies, their integration in AUTOMSv2 does not influence its performance since, as already stated, the profiling analysis automatically detects the features of the input ontologies and exclude these methods from computing alignments (i.e. are not included in the synthesis configuration for the smart/control entities' ontology definitions).

The different configurations regarding the way the above methods were synthesized, i.e. computing and synthesizing alignments, is based on the profiling information gathered after the analysis of the input ontologies. Both input ontologies (since our problem concerns the alignment of two ontologies), are examined using different analysis methods, as the example following ones:

1. Based on the size of the ontologies, i.e. the number of classes that ontologies have, if one of them has more than a specific number of classes (this number is experimentally set to 100), then this pair of ontologies is not provided as input to alignment methods with heavy computations since it will compromise the overall execution time of the tool. Such methods are the vector-based, WordNet-based and structure-based ones.
2. If an ontology pair contains an ontology with no instances at all, then this pair is not provided as input to any instance-based alignment method (the explanation for this is straight forward).
3. If an ontology pair contains two ontologies that a specific number of their entities have no names with an entry in WordNet, but they have labels, then provide this pair as input to alignment methods that a) do not consider WordNet as an external resource and b) consider labels matching instead of class names.
4. If an ontology pair contains two ontologies that a specific number of their entities have no names with an entry in WordNet, and they also have no labels, then provide this pair as input to alignment methods that a) do not consider WordNet as an external resource and b) do not consider labels' matching.

AUTOMSv2 is using a free Java API named WebTranslator (http://webtranslator.sourceforge.net/) in order to solve the multi-language problem. AUTOMSv2 translation method is converting the labels of classes and properties that are found to be in a non-English language (any language that WebTranslator supports) and creates a copy of an English-labeled ontology file for each non-English ontology. This process is performed before AUTOMSv2 profiling, configuration and matching methods are executed, so their input will consider only English-labeled copies of ontologies.

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

AUTOMSv2 web page (short description, the system and OAEI results) is currently hosted at http://ai-lab-webserver.aegean.gr/kotis/AUTOMSv2.

2 Results

In this paper we conjecture that we must also shortly present a snapshot of AUTOMSv2 participation in 2011.5 campaign. This was motivated by the capability of giving a rough comparison with other tools also participated in the same contest, and also comparing it with latest versions of our own tools that participated in the OAEI 2012 contest. A pre-final experimental version of AUTOMSv2 was submitted in 18th of March 2012 as a submission to the Ontology Alignment Evaluation Initiative 2011.5 Campaign (http://oaei.ontologymatching.org/2011.5/seals-eval.html), using the Semantic Evaluation At Large Scale (SEALS) platform.

The Benchmark results ("biblio" dataset) for OAEI 2011.5 (http://oaei.ontologymatching.org/2011.5/results/benchmarks/index.html) indicated that AUTOMSv2 could perform quite high in terms of precision (0.97) and low for recall (0.54). Its f-measure (0.69) was the 6th best in 14 tools participated (only for this particular dataset). In terms of runtime measurements, AUTOMSv2 was placed in the 8th place in 13 tools, which was not an expecting result due to the profiling and configuration optimization strategy the AUTOMSv2 follows.

The Conference results for OAEI 2011.5 (http://oaei.ontologymatching.org/2011.5/results/conference/index.html) again indicated that AUTOMSv2 could perform quite higher in terms of precision (0.75 and 0.79) and lower for recall (0.4 and 0.43), where the highest precision of other tools was 0.78 and 0.82. In terms of runtime performance AUTOMSv2 performed quite similar to Benchmark results.

The Multifarm results for OAEI 2011.5 (http://oaei.ontologymatching.org/2011.5/results/multifarm/index.html) indicated that AUTOMSv2 could perform quite well with multilingual ontologies, obtained the 2nd better f-measure result (0.36) among 12 tools (for type I dataset – different ontologies), with an average precision of 0.63 and a recall of 0.25.

For Anatomy and Large Biomedical Ontologies tracks of OAEI 2011.5, AUTOMSv2 did not generate any results.

2.1 Benchmark 2012

The Benchmark results for OAEI 2012 (http://oaei.ontologymatching.org/2012/benchmarks/index.html) indicated that AUTOMSv2 could perform quite high in terms of precision (range between 0.91 and 0.99) and low for recall (range between 0.51 and 0.55) for the four out of five domains (see Table 1). For the last domain, i.e. finance, the tool performed similarly in terms of recall (0.55) but unexpectedly (blind test) in terms of precision (0.35). Comparing to 2011.5 results, AUTOMSv2 has not improved its performance.

Table 1. Scores for Benchmark track 2012

	Precision	F-measure	Recall	Runtime(s):
biblio	0.97	0.69	0.54	58
benchmark-2	0.97	0.68	0.52	161
benchmark-3	0.99	0.7	0.54	519
benchmark-4	0.91	0.65	0.51	421
finance	0.35	0.42	0.55	1535

2.2 Conference 2012

The Conference results for OAEI 2012 (http://oaei.ontologymatching.org/2012/conference/index.html) indicated that AUTOMSv2 could perform higher in terms of precision (range between 0.64 and 0.67) and lower for recall (range between 0.33 and 0.36).

AUTOMSv2 failed to generate 6 alignments out of 120 test cases. An improved version delivered after deadline succeeded to generate all alignments 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. Having said that, improved version delivered after deadline succeeded to generate all alignments with improved performance (in the case of ra1: Precision=0.79, F1-measure=0.56, Recall=0.43 and in the case of ra2: Precision=0.75, F1-measure=0.52, Recall=0.4)

Table 2. Scores for Conference track 2012

Official (before deadline)				
	Precision	F-measure	Recall	Runtime(ms)
r1	0.67	0.47	0.36	452477
r2	0.64	0.44	0.33	452477
Improved (after deadline)				
	Precision	F-Measure	Recall	Runtime
r1	0.79	0.56	0.43	same
r2	0.75	0.52	0.4	same

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.

Comparing to 2011.5 results, AUTOMSv2 has not improved its performance (compared with the official results).

2.3 Multifarm 2012

The Multifarm results for OAEI 2012 (http://www.irit.fr/OAEI/) indicated that AUTOMSv2 could perform for all pairs apart from the ones involving Czech, Russian and Chinese.

Table 3. Scores for Multifarm track 2012

Official (before deadline)				
	Precision	F-measure	Recall	Runtime(s)
de-en	0.91	0.35	0.22	891
de-es	0.82	0.26	0.15	1752
de-fr	0.93	0.25	0.14	1842
de-nl	0.88	0.31	0.19	1694
de-pt	0.9	0.25	0.15	1714
en-es	0.71	0.32	0.21	886
en-fr	0.75	0.32	0.2	1006
en-nl	0.78	0.35	0.23	851
en-pt	0.75	0.29	0.18	926
es-fr	0.74	0.29	0.18	1668
es-nl	0.7	0.34	0.22	1757
es-pt	0.7	0.36	0.25	1748
fr-nl	0.71	0.26	0.16	1735
fr-pt	0.74	0.26	0.16	1699
Average	0.79	0.30	0.19	1441

For the non-zero computed pairs, the tool performed higher in terms of precision (range between 0.7 and 0.91) and lower for recall (range between 0.14 and 0.25). In this paper we decided to present results (see Table 3) 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 that we received from organizers of this track also.

Comparing to 2011.5 results, AUTOMSv2 has not improved its performance. In fact, the f-measure has been decreased by 0.6. Comparing the average results of precision and recall between the two contests, we can observe that the average precision was increased while the average recall was decreased significantly.

2.4 LargeBio 2012

The LargeBio results for OAEI 2012 indicated that AUTOMSv2 could perform also with large datasets, although with large runtimes (17 hours). The results are depicted in Table 4. As expected, AUTOMSv2 could perform higher in terms of precision (range between 0.79 and 0.82) and lower for recall (range between 0.49 and 0.52).

Table 4. Scores for LargeBio track 2012

FMA-NCI	Precision	Recall
Original UMLS mappings	0.82	0.49
Refined UMLS mappings using LogMap's repair facility	0.80	0.50
Refined UMLS mappings using Alcomo debugging system	0.79	0.51
Harmonized mapping set from OAEI 2011.5	0.82	0.52

3 Comments

As already stated, the aim of this development experience was not to deliver a tool to compete with others in terms of precision and recall. Instead, we aimed at the development of a new version of AUTOMS (Automating the Synthesis of Ontology Mapping Methods) using new and state-of-the-art technologies and alignment methods. Nevertheless, AUTOMSv2 obtained good (above average) results both in OAEI 2011.5 and 2012 contests.

The following table summarizes the features of ASE tool:

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

AUTOMSv2 results could have been better and computation of results could have been performed for other tracks (Library, Anatomy). We experienced a lot of unexpected difficulties with bugs appeared in third-party libraries such as in Alignment API, COCLU string similarity method, WebTranslator API, Microsoft Bing Translator API.

Our future plans to integrate also the computation of subsumption relation between concepts/properties has been lately realized in a new tool called ASE (Aligning Smart Entities), also participating in this contest as a first prototype version. Also, we plan to optimize the performance of our ontology alignment tools by adapting the configurations of the synthesized methods in a more efficient manner.

4 Conclusion

This paper presented AUTOMSv2 tool and evaluation results obtained for OAEI 2011.5 and 2012 contests. This effort was the result of our motivation to rebuild AUTOMS by putting together a) a well-known, widely used and continuously evolving/maintained alignment framework b) the synthesis of state-of-the-art alignment methods, c) a modern approach of synthesizing methods using profiling and configuration strategies, and d) multilingual support. Although our aim was not to compete with other tools in precision and recall, nevertheless, AUTOMSv2 obtained good results that we have also compared with results of other tools obtained for OAEI 2011 and 2011.5 contests.

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)